UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMATICA

Departamento de Arquitectura de Computadores y Automática

MINERÍA DE TEXTO APLICADA A BIOINFORMÁTICA FUNCIONAL

MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR

Mariana Lara Neves

Bajo la dirección del doctor

Alberto Pascual Montano

Madrid, 2013

Minería de Texto aplicada a

Bioinformática Funcional

Departamento de Arquitectura de Computadores y Automática

Facultad de Informática

Universidad Complutense de Madrid

TESIS DOCTORAL

Mariana Lara Neves

Director: D. Alberto Pascual Montano

Madrid, Junio de 2012

Minería de Texto aplicada a

Bioinformática Funcional

Mariana Lara Neves

Director: D. Alberto Pascual Montano

Ao meus pais,

ao meu marido Julián

e à minha filha Nicole

Agradecimientos

Me gustaría empezar la serie de agradecimientos con el director de mi tesis Alberto por darme la oportunidad de trabajar en lo que yo siempre había deseado y a José María Carazo por acogerme en su grupo durante mi estancia en el CNB.

Luego me gustaría agradecer a mis compañeros de la Complutense, en especial Rubén que siempre me ha apoyado en mis momentos difíciles con la tesis, además de ser un gran amigo. Y por haberme ayudado en todo el papeleo que conlleva el trámite de una tesis doctoral, ya que desde Berlín no he podido hacerlo. Luego agradecer también a Miguel y a Edgardo, mis dos otros compañeros de doctorado, así como a los técnicos del laboratorio del ArTeCS: Enrique, Cesar, Adrián y Luis. Muchas gracias por haberme ayudado tantas veces con mis problemas con Marbore.

También agradezco a mis compañeros del CNB, en especial Natalia, que además de haberse convertido en una gran amiga y compañera de mesa, siempre me apoyó cuando me surgían dudas sobre si sería capaz de terminar la tesis. También me aportó mucho dándome valiosos consejos en otra faceta de la vida, como es el embarazo de una madre primeriza. Y a Mónica, que también me ha ayudado alguna que otra vez con determinadas dudas de "Text Mining". Luego agradezco a Jesús por todo el apoyo técnico, siempre muy paciente, contestado a todas nuestras preguntas y solucionando nuestros problemas con los portátiles y los servidores. Muchas gracias también a Blanca, nuestra secretaria, por encargarse con tanta paciencia de nuestros contratos, viajes, cafeteras, regalos, celebraciones, etc. Y gracias a todos mis compañeros del CNB, los que se han ido y los que siguen: Adrián, Analu, Carlos Oscar, Carmen, Federico, Javi, Joan, Joaquín, José Miguel, José Ramón, Johannes, Juanjo, Marabini, Melero, Marta, Rocío, Román, Sami, Sjors y Vital.

Por fin agradezco a mi marido Julián, que me ha apoyado en todo estos últimos años, siempre insistiendo que poquito a poquito terminaría la tesis, despacito pero constante. Y a mi hija Nicole, que aunque con sus poco más de año y medio no se entere de mucha cosa, nos ha dado muchas alegrías y más motivos para seguir luchando por nuestros objetivos y nuestros nuevos retos. Su primer cumpleaños me sirvió como "deadline" para la primera versión de esta tesis. Y su segundo cumpleaños me servirá como "deadline" para lectura de la misma. Finalmente, a mis padres y hermanos en Brasil, que desde lejos me han apoyado para que terminara la tesis y me han inculcado desde pequeña la importancia de seguir estudiando siempre.

TABLE OF CONTENTS

Resumen

Esta tesis doctoral propone metodologías para solucionar diversos problemas en el campo de la minería de datos biomédica. Aunque puedan parecer que no están conectados cuando descritos individualmente, en su conjunto, consisten en pasos necesarios para la automatización de los procesos de extracción automática de conocimiento a partir de la literatura biomédica. Más específicamente, esta tesis trata de las tareas de reconocimiento de entidades biológicas, extracción de relaciones y normalización de menciones.

Las metodologías que proponemos para las tareas de reconocimiento de entidades biológicas y extracción de relaciones utiliza el razonamiento basado en casos, parte del paradigma de aprendizaje automático. Para la extracción de entidades biológicas, mas específicamente de genes/proteínas, evaluamos los métodos con los datos disponibles en la competición de BioCreative II. Como resultado de este trabajo, hemos desarrollado una librería Java llamada Moara, que incluye la implementación de estos métodos y la posibilidad de entrenarlos con datos distintos del utilizado en el desarrollo del sistema. Nuestros métodos también han sido integrados a la plataforma U-Compare, que permite la utilización inmediata de nuestros métodos además de su comparación con otros sistemas.

También hemos aplicado el razonamiento basado en casos para la extracción de eventos biológicos, que consiste en un previo reconocimiento de los términos claves de un evento (e.g., "expresión"), seguido de la extracción de sus argumentos (e.g., proteína, localización). Nuestros métodos han sido evaluados con los datos disponibles en la competición de BioNLP 2009 Event Extraction, en la que ha participado una primera versión del sistema. Esta metodología también ha sido integrada a la plataforma U-Compare como parte de un servidor que incluye algunos de los participantes de la competición. También hemos realizado experimentos con el corpus de BioText para la extracción de asociaciones entre enfermedades y tratamientos, como forma a demonstrar que nuestros métodos también se comportan satisfactoriamente para otros tipos de relaciones biomédicas.

Finalmente, hemos propuesto metodologías para la normalización de menciones de genes/proteínas. Nuestros métodos están basados en una comparación exacta de un diccionario de sinónimos con las menciones y en algoritmos de aprendizaje automático, además de la desambiguación de los identificadores. Una evaluación y los resultados son presentados para cada una de las metodologías utilizando los datos disponibles en las dos primeras ediciones de la competición BioCreative. Los documentos se refieren a cuatro organismos: humano, ratón, mosca y levadura. Los métodos desarrollados para la normalización de genes/proteínas también están incluidos en la librería Java de Moara, que además permite el entrenamiento del sistema con nuevos organismos.

Abstract

This thesis describes new methodologies proposed to solve several state of the art biomedical text mining problems, that when described individually seems unconnected but altogether represent necessary steps in the automatic process to extract knowledge from biomedical literature. In particular the thesis is focused on the tasks of named entity recognition, relationship extraction and entity mention normalization.

The methodologies we propose for the tasks of recognition of entities and relationship extraction use the case-based reasoning approach, which is part of the machine learning paradigm. For the named-entity recognition task, we apply these methods for the extraction of genes and proteins, which are evaluated using the BioCreative II Gene Mention corpus. As result of this work, we have developed the Moara Java library, which include the implementation of our methods and the possibility of training them with extra corpora. Our methods have also been integrated into the U-Compare framework, which allow their instant use and the comparison to other systems.

We also apply case-based reasoning for the biomedical events extraction, which include first the recognition of the event triggers (e.g., “expression”, “regulates”), followed by the extraction of the arguments which compose the event, such as theme, cause and location. For the extraction of biomedical events, the methods were evaluated using the datasets available for the BioNLP 2009 Event Extraction challenge and our first approach participated in the competition. Our methodology has also been integrated into the U-Compare framework as part of a meta-server with some of the participants of the challenge. Finally, we have also carried out experiments with the BioText corpus for the extraction of relationships between diseases and treatments, in order to prove that our methods also perform satisfactorily for a different type of relationship.

We also propose methodologies for the normalization of the genes and proteins mentions to their identifiers. Our approaches are based on dictionary lookup and machine learning algorithms and they include the disambiguation of the identifiers. Evaluation and results are also presented for each approach using datasets available in two of the BioCreative challenges for four organisms: yeast, mouse, fly and human. The methods developed for the normalization of gene mentions are also included in the Moara Java library, adding the possibility of training them with extra organisms.

Abbreviations

AL atLoc (biological event's argument)

BIN Binding (biological event)

CA Cause (biological event's argument)

CAT Protein catabolism (biological event)

CBR Case-based reasoning

CbrBC2 CBR-Tagger model based on the BioCreative 2 Gene Mention dataset

CbrBC2y CBR-Tagger model based on the BioCreative 2 Gene Mention and on the BioCreative task 1B for the yeast datasets

CbrBC2m CBR-Tagger model based on the BioCreative 2 Gene Mention and on the BioCreative task 1B for the mouse datasets

CbrBC2f CBR-Tagger model based on the BioCreative 2 Gene Mention and on the BioCreative task 1B for the fly datasets

CbrBC2ymf CBR-Tagger model based on the BioCreative 2 Gene Mention and on the BioCreative task 1B for the yeast, mouse and fly datasets

CS CSite (biological event's argument)

DNA Deoxyribonucleic acid

EMN Entity mention normalization

EXP gene expression (biological event)

FM F-Measure

FN False negative

FP False positive

IE Information extraction

LCS Least Common Subsumer

LOC Localization (biological event)

MFC Minimum Frequency of the Case

MMF Minimum Matching Feature

NEG Negative Regulation (biological event)

NER Named Entity Recognition

NLP Natural Language Processing

P Precision

PHO Phosphorylation (biological event)

POS Part-of-Speech

POS Positive Regulation (biological event)

PPI Protein-protein interaction

R Recall

REG Regulation (biological event)

RNA Ribonucleic acid

ST, ST2, ST3 Site, Site2, Site3 (biological event's arguments)

TH, TH2, TH3 Theme, Theme2, Theme3 (biological event's arguments)

TL toLoc (biological event's argument)

TM Text mining

TP True positive

TRA Transcription (biological event)

WWW World Wide Web

List of Figures

List of Tables

CHAPTER 1 RESUMEN EN ESPAÑOL

1.1 Introducción y Motivación

La biología molecular es la disciplina que estudia la biología a nivel molecular (Lodish, Berk et al. 2000), es decir, los procesos importantes relacionados con los seres vivos, tales como la estructura, función y composición molecular. Esta disciplina se relaciona con las Ciencias Biológicas y la Química, y más particularmente con la Bioquímica y con la Genética, que se refiere a la comprensión de los diversos sistemas de la célula, tales como el ácido desoxirribonucleico (ADN), el ácido ribonucleico (ARN), la biosíntesis de proteínas, el metabolismo y la forma en que estas interacciones son reguladas con el fin de lograr un funcionamiento correcto de la célula.

En cuanto a otros campos relacionados con la biología molecular, la bioquímica se centra más en el rol, función y estructura de las biomoléculas. Estas últimas pueden ser definidas como cualquier molécula orgánica que puede ser producida por un ser vivo, algunas de grande dimensiones, como las proteínas y los ácidos nucleicos, y algunas más pequeñas, tal como los metabolitos. Por otro lado, la genética puede ser definida como el estudio de la diferencia entre los diversos organismos debido, por ejemplo, a la ausencia de un gen. La genética incluye también el estudio de los mutantes, es decir, organismos que carece de un o más componentes en relación con el fenotipo normal, cualquier característica observable de un organismo, tal como su morfología, desarrollo, comportamiento, etc. Por último, la Biología Molecular incluye el estudio de los procesos de replicación del ADN, transcripción (o la síntesis de ARN) y la traducción del ARN a aminoácidos.

Los avances en las tecnologías biomédicas y los experimentos de alto rendimiento, tales como los microarrays de ADN, expresión génica (Holloway, van Laar et al. 2002), técnicas de espectrometría de masa proteómica (MacBeath 2002), la secuenciación de próxima generación, entre otros, están siendo ampliamente utilizados desde la última década y han permitido que los científicos estudien los sistemas biológicos desde una perspectiva global. Estas nuevas metodologías generan grandes cantidades de información relacionadas con genes y proteínas a diferentes niveles. Por lo tanto, el reto radica en la capacidad de analizar e interpretar estos datos, en el que la comunidad bioinformática ha realizado importantes avances. Sin embargo, estas tareas no son triviales y el desarrollo de métodos automáticos son necesarios con el fin de facilitar la interpretación funcional y extraer hechos interpretables y conocimiento biológico. Este es uno de los principales retos de la investigación en bioinformática.

La comprensión de los complejos biológicos complejos en organismos eucariotas, como los humanos, inevitablemente necesita la integración de todos los posibles datos experimentales provenientes de estudios de algunos procesos particulares en distintos organismos. La integración de esta información requiere una precisa

interpretación y análisis de muchas fuentes de información. Actualmente, no hay una base de conocimientos única en la que esta información pueda ser completamente encontrada en una forma estructurada. Por otro lado, la literatura científica es probablemente una de las fuentes más ricas de información. Por esta razón, desarrollar métodos para la extracción y la organización de esta información de forma automática para permitir su posterior análisis es un gran desafío para la comunidad científica.

En la última década, el interés por la minería de textos biomédicos de los textos clínicos y la literatura científica ha experimentado un enorme incremento (Chapman and Cohen 2009). La razón principal es que la literatura abarca casi todos los aspectos de la biología, química y medicina, donde apenas hay límites al tipo de información que puede ser recuperada a través del uso de una exhaustiva y cuidadosa minería de textos. En este dominio, la minería de texto biomédica puede definirse como el conjunto de métodos utilizados para extraer o recuperar el conocimiento que se oculta en los textos y presentarlo de una manera coherente para que pueda ser utilizado por especialistas en ciencias de la vida. Por lo tanto, la minería de texto se encarga de analizar los textos con el fin de descubrir nueva información que sería difícil de ser recuperada de otra forma.

El creciente interés en la minería de textos biomédicos se relaciona con el crecimiento y acumulación de la literatura científica y el rápido proceso de descubrimiento biomédico de información. Los métodos de cálculo utilizados para el procesamiento de la literatura biomédica permite el acceso fácil y rápido de los biólogos, bioinformáticos y anotadores (curators) de base de datos a los textos pertinentes. Sin embargo, generalmente, la tarea de extracción de información se lleva a cabo manualmente. La información se extrae de las publicaciones científicas pertinentes y se la almacena en enormes bases de datos, que están por lo general disponibles libremente a la comunidad científica, tales como EntrezGene (Maglott, Ostell et al. 2007) y Uniprot (2009) para información sobre el genoma, y MINT (Chatr-aryamontri, Ceol et al. 2007) y IntAct (Kerrien, Alam-Faruque et al. 2007) en el dominio de la interacción de proteínas. Estas bases de datos son de gran importancia ya que los resultados de experimentos y de distintos métodos de la bioinformática suelen interpretarse mediante el uso de la información que contienen. Hoy en día, es inconcebible intentar entender un proceso biológico complejo en algunas condiciones determinadas en organismos complejos, como el humano, sin tener en cuenta toda la información que se podría resumir de procesos similares llevados a cabo por la comunidad científica.

Desafortunadamente, toda esta información no siempre se encuentra en un formato estructurado tal como las bases de datos relacionales, que son de fácil acceso a los investigadores. Por el contrario, la mayor parte de esta información se encuentra en un formato no estructurado en textos o documentos mal estructurados, tales como

PubMed (Fenton and Williams 2005). Debido a esta situación, un gran número de grupos de investigación en las áreas de la Biología Molecular y la Informática han dedicado grandes esfuerzos en el desarrollo de nuevas metodologías para la extracción de grandes cantidades de información de la literatura científica. Este es el alcance del trabajo descrito en esta tesis.

Con el fin de procesar y almacenar esta información, muchos son los métodos de cálculo que se han propuesto en las áreas de bioinformática, biología computacional y ciencias de la computación, y en especial, en el ámbito del procesamiento del lenguaje natural (PLN). Este último puede definirse como el conjunto de métodos para el tratamiento automático de documentos escritos en lenguaje natural, como en la lengua inglesa. Un sistema clásico y completo de minería de textos puede estar compuesto de varios módulos siendo los más comunes los más comunes la recuperación de información o documentos importantes, el reconocimiento de entidades, extracción de información y el descubrimiento de conocimiento (Jackson and Moulinier 2002).

La recuperación de la información (Hersh 2008) se encarga de la búsqueda y recuperación de documentos que coincidan con alguna consulta a una gran base de documentos, tal como la World Wide Web (WWW). En el ámbito biomédico, ella puede ser descrita como una forma de reunir los textos pertinentes, generalmente documentos científicos, a partir de la base de datos de PubMed (Fenton and Williams 2005). Por lo general, este es el primer paso en cualquier sistema de extracción de texto. La recuperación de la información también se puede realizar dentro de un texto con el fin de decidir qué partes de un documento es más relevante de acuerdo con la consulta del usuario.

El reconocimiento de entidades (Park and Kim 2006) es la identificación de algunas entidades predefinidas en un determinado texto, que pueden haber sido adquirido por medio de la recuperación de información o de cualquier otra forma, tal como manualmente. En el ámbito biomédico, las entidades suelen ser genes, proteínas, enfermedades, fármacos, partes anatómicas, etc. Es muy común la extracción de más de un tipo de entidad a partir de un texto a la vez, y una categorización puede ser necesaria con el fin de definir el tipo de entidad para cada una de las menciones se extrae. Aún relacionado a este paso, la normalización de entidades está definida como la asociación de cada mención de un nombre a un identificador en una ontología o terminología predefinida. Este paso puede realizarse junto con el reconocimiento de entidades o separadamente.

La extracción de información se encarga de extraer relaciones predefinidas entre algunas entidades a partir del texto de un documento no estructurado. Las entidades deben haber sido previamente extraídas en la etapa de reconocimiento de entidades. En el dominio biomédico, las interacciones entre proteínas (Krallinger,

Leitner et al. 2008) es una de las tareas más populares de extracción de información. La importancia de la extracción de información va en aumento debido al creciente interés en la biología de sistemas (Ananiadou, Pyysalo et al.).

Finalmente, el descubrimiento de conocimiento intenta encontrar la información oculta o implícita en los textos. Esto se realiza generalmente mediante la propuesta de algunas hipótesis posibles derivadas de la información extraída en el paso anterior. Por ejemplo, de forma a inferir relaciones indirectas, un sistema de descubrimiento de conocimiento puede generar una hipótesis de que A está relacionado con C si el texto describe que A se relaciona con B y B con C.

1.2 Objetivos

Esta tesis propone nuevas metodologías para resolver varios problemas de la minería de texto biomédica. Cuando se describen de forma individual, estas tareas parecen estar desconectadas, pero en conjunto, son pasos necesarios en el proceso automático de la extracción de conocimiento de la literatura biomédica. En particular, esta tesis se centra en el reconocimiento de entidades, la extracción de relaciones y la normalización de menciones de entidades. Detalles de nuestra contribución y un esbozo de la tesis se describen a continuación.

El capítulo 2 comienza con una introducción al procesamiento del lenguaje natural, tal como la tokenización, el etiquetado gramatical y el análisis sintáctico. Estos serán seguidos por las métricas utilizadas para la evaluación de las metodologías que aquí se propone. Finalmente, se describirá el estado del arte de las siguientes tareas: el reconocimiento y la normalización de genes y proteínas y la extracción de las relaciones biomédicas.

El capítulo 3 describe en detalle las metodologías que se proponen para las tareas de reconocimiento de entidades, y particularmente para la extracción de genes y proteínas y los disparadores (triggers) de eventos biológicos. También se describirán los métodos para la extracción de información, tales como eventos biomédicos y relaciones entre tratamientos y enfermedades. Las metodologías que proponemos utiliza el abordaje de razonamiento basado en casos. Una evaluación y los resultados serán presentados para cada una de esas tareas.

El capítulo 4 describe los métodos para la normalización de los genes y proteínas respeto a sus identificadores. Nuestros métodos se basan en la consulta de diccionarios y algoritmos de aprendizaje automático e incluyen la desambiguación de los identificadores, cuando sea necesario. Una evaluación y resultados también serán presentados para cada abordaje.

El capítulo 5 presenta las conclusiones de esta tesis y discutirá los trabajos futuros que están en curso o que tenemos la intención de llevar a cabo para la continuidad de la las metodologías y las aplicaciones que se han desarrollado durante la tesis.

El desarrollo de nuevos métodos para la recuperación de la información y el descubrimiento de conocimiento están fuera del alcance de este trabajo y no son tratados en la tesis. En lugar de realizar la recuperación de documentos relevantes, se han utilizado corpora estándares en las evaluaciones de cada uno de las metodologías que aquí se propone.

En el apéndice se describen los materiales que han sido utilizados para el desarrollo de los algoritmos y su evaluación. Se incluye bases de datos y terminologías, detalles sobre el corpus utilizados en la evaluación y una breve descripción de las bibliotecas externas que se han utilizado en el desarrollo de nuestros sistemas y metodologías. El apéndice también incluye las funcionalidades de nuestro sistema que están disponibles para su uso como parte del proyecto Moara: CBR-Tagger, ML-Normalización y BioEvent Extractor. Finalmente, se presenta el listado de las publicaciones que están relacionadas con esta tesis, tanto en revistas como en conferencias y workshops.

1.3 Estado del Arte

1.3.1 Reconocimiento de genes y proteínas

El reconocimiento de entidades en los documentos de texto es la identificación de algunos tipos predefinidos de entidades, tal como personas, organizaciones o lugares (Park and Kim 2006). Como resultado, una entidad (o una mención) puede definirse como una frase compuesta de una o más palabras que denotan un objeto específico, tal como una persona, organización o ciudad. En el ámbito biomédico, estas entidades son por lo general genes, proteínas, enfermedades, líneas celulares, etc. La extracción de estas entidades es un paso previo para otras tareas de minería de texto, tales como la recuperación de documentos relacionados con un determinado gen o proteína, o la caracterización inicial de las proteínas implicadas en un texto para un sistema de extracción de interacciones entre proteínas o de procesos de expresión génica, por ejemplo. Esta tarea incluye la identificación exacta de la mención en el texto, es decir, de sus límites, que se definen por la posición del primer y último caracteres que delimitan las palabras que lo componen.

La principal dificultad relativa al reconocimiento de genes y proteínas es la existencia de un gran número de estas entidades, la falta de normativas relativas a su nomenclatura y la resistencia de la comunidad científica para utilizar las nomenclaturas existentes (Tamames and Valencia 2006). La nomenclatura de

genes y proteínas sufre de varios problemas, tales como la ambigüedad, sinónimos y variantes.

En cuanto a la ambigüedad, entidades diferentes pueden compartir el mismo nombre y algunos nombres incluso puede coincidir con palabras comunes en inglés (por ejemplo, "deafness"), lo que complica aún más su detección en un texto (Leser and Hakenberg 2005). Además, a las entidades recién descubiertas, a veces se les asigna un nombre que es ya está en uso por otro gen o proteína existente.

Además, un gen o una proteína pueden tener más de un nombre, normalmente llamados sinónimos o alias. Este problema hace que la construcción de un listado completo de los sinónimos para un determinado organismo sea una tarea mucho más difícil, incluso con la ayuda de expertos. A veces, ni siquiera las bases de datos específicas para un determinado organismo son capaces de mantener un listado complejo y dinámico de sinónimos.

Finalmente, las variaciones también son muy comunes en la nomenclatura de genes y proteínas, tales como: (1) a nivel de carácter: la presencia o ausencia de caracteres especiales, como guiones, paréntesis, mayúsculas/minúsculas, comas, etc.; (2) a nivel de palabra: debido al uso de sinónimos como parte del nombre, tales como "gastric" y "stomach"; (3) en el orden de las palabras: cuando las mismas palabras aparecen en un orden diferente; (4) las abreviaciones, por ejemplo, el uso de "TNF" en lugar de "tumor necrosis factor".

La tarea de reconocimiento de genes y proteínas se utiliza a menudo como un preludio de otros problemas, tales como la normalización de genes y proteínas. Este último se convierte en una tarea más fácil si se proporciona las menciones disponibles en el texto, así como su localización exacta en el mismo. Además, el uso de información de contexto, es decir, las palabras o frases cerca de la mención, puede facilitar la tarea de normalización (cf. apartado 2.4).

Los abordajes propuestos para el reconocimiento de genes y proteínas pueden clasificarse en métodos basados en un diccionario de sinónimos, reglas manuales y aprendizaje de máquina. Debido al tiempo que se tarda en registrar un nuevo sinónimo para un gen o una proteína, métodos que se basan solo en diccionarios no son capaces de reconocer sinónimos recientes. Además, las variaciones en la nomenclatura no siempre se incluyen en estos listados. Por lo tanto, algunas de las metodologías más exitosas (Hanisch, Fundel et al. 2005) se basan en un listado inicial de sinónimos que son expandidos de forma manual y automática con algunas variaciones.

Muchas soluciones han sido propuestas para la tarea de reconocimiento de genes y proteínas (Smith, Tanabe et al. 2008). Los métodos van desde sistemas basados en

reglas (Fukuda, Tsunoda et al. 1998) a similitud de palabras (Krauthammer, Rzhetsky et al. 2000). La gran mayoría de los sistemas que han obtenido buenos resultados en esta tarea han utilizado el algoritmo de Condicional Random Fields (CRF) (Lafferty, McCallum et al. 2001), un método de probabilidad condicional ampliamente utilizado para la clasificación de datos. Otros ejemplos de sistemas basados en esta metodología son los trabajos de (Dai, Hung et al. 2007) y (Katrenko and Adriaans 2007), así como Abner (Settles 2005) y BANNER (Leaman and González 2008), dos de las herramientas de reconocimiento de genes y proteínas más ampliamente utilizadas.

Otras metodologías incluyen inferencia bidireccional utilizada para el desarrollo del sistema GENIA (Tsuruoka and Tsujii 2005), Support Vector machines (SVM) en el trabajo de (Chen, Liu et al. 2007) y (Huang, Lin et al. 2007), aprendizaje semi supervisionado (Ando 2007), que obtuvo los mejores resultados en la competición de BioCreative (Smith, Tanabe et al. 2008), razonamiento basado en casos (Neves, Carazo et al. 2010) y un conjunto de algoritmos de aprendizaje de máquina (Naïve Bayes, k-NN, AdaBoost con Naïve Bayes y árboles de decisión C4.5) (García, Puertas et al. 2007). Además, el trabajo de (Zhou, Shen et al. 2005) utiliza varios métodos y los unifica utilizando una sistema de retroalimentación.

Al utilizar algoritmos de aprendizaje automático para el reconocimiento automático de genes y proteínas, algunas características del texto deben de ser proporcionada. Algunos ejemplos de tales características se enumeran a continuación: longitud de la mención (Huang, Lin et al 2007); indicativo de si la mención es el nombre completo o una abreviatura; la posición de la palabra en la frase (García, Puertas et al. 2007); atributos ortográficos (Klinger, Friedrich et al. 2007), como mayúsculas y minúsculas (en distintas posiciones en la mención); la presencia de caracteres especiales (paréntesis, guiones, etc.); números (enteros, reales o romanos); letras griegas (Neves 2007); indicativo de si la referencia se produce dentro de comillas o corchetes (Dai Hung et al. 2007); raíz o lema de la mención y de las palabras vecinas (Vlachos 2007); etiquetas gramaticales (Klinger, Friedrich et al. 2007) (Finkel, Dingare et al. 2005); sufijos y prefijos compuestos de dos a cuatro caracteres para la identificación de las palabras del entorno biológico (Chen, Liu et al. 2007); n-grams, por lo general bigrams o trigrams (Struble, Povinelli et al. 2007); presencia de determinados términos biomédicos (Struble, Povinelli et al. 2007) (Ganchev, Crammer et al. 2007); presencia de los símbolos de tres letras que representan los aminoácidos o nucleótidos (Kuo, Chang et al. 2007); clasificación de las menciones según algunas clases predefinidas (Tamames 2005); y cualquiera de las características descritas anteriormente para una ventana de contexto delimitado (Huang, Lin et al. 2007), es decir, un cierto número predefinido de palabras que viene antes o después de la mención.

1.3.2 Extracción de relaciones biomédicas

En el procesamiento de lenguaje natural, la extracción de información puede definirse como la tarea de encontrar una información específica en un documento de texto. Esta información puede estar relacionada con una o más entidades predefinidas (Ananiadou and Nenadic 2006). Más específicamente, la extracción de relaciones biomédicas es una especialización de la tarea de extracción de información en que las entidades están relacionadas con los dominios de la medicina o de la biología molecular, tales como los genes, proteínas o enfermedades.

La extracción de relaciones es un tema clave en la minería de texto, ya que toma parte en muchos procesos biológicos, y muchos esfuerzos han sido dedicados a este asunto. Como ejemplo, bases de datos están disponibles para el almacenamiento de interacciones entre pares de proteínas, tales como MINT (chatr-aryamontri, Ceol et al. 2007) y IntAct (Kerrien, Alam-Faruque et al. 2007). La mayor parte de los datos contenidos en estas bases de datos han sido extraídos manualmente por expertos.

Por el momento, la tarea más popular en la minería de textos biomédicos es la interacción entre proteínas (Krallinger, Leitner et al. 2008) y más recientemente, la extracción de eventos biomédicos (Kim, Ohta et al. 2009), ambos debido a competiciones que se han producido en los últimos años. Estos desafíos, y por consiguiente la disponibilidad de corpora anotados, han aumentado el número de soluciones para la extracción de las relaciones biomédicas, así como la mejora los resultados. Por ejemplo, el BioCreative II protein-protein interaction task (Krallinger, Leitner et al. 2008) consistió en cuatro tareas, incluyendo la extracción de esas interacciones en el texto completo de publicaciones.

Más recientemente, el BioNLP Shared Task Event Extraction (Kim, Ohta et al. 2009) ha propuesto la identificación de una variedad de eventos biomédicos. Algunos de los eventos eran más fáciles de extraer, tales como la expresión génica o la fosforilación, mientras que otros eran más complejos, tales como binding y regulación génica. Además, la identificación de negaciones y especulaciones relacionados al evento han hecho con que las tareas fueran aún más complejas. El F-measure de los mejores participantes ha variado desde aproximadamente 40% (evento de regulación negativa) a casi el 80% (expresión génica).

En cuanto a los corpora anotados con las relaciones biomédicas, citamos iniciativas tales como GENIA (Kim, Ohta et al. 2008), GREC (Thompson, Iqbal et al. 2009), BioCreative corpus PPI (Krallinger, Leitner et al. 2008) y el formato unificado de cinco corpora de interacciones entre proteínas (Pyysalo, Airola et al. 2008). Estos corpora han permitido el desarrollo de muchas soluciones basadas en algoritmos de aprendizaje supervisado (Tikk, Thomas, et al. 2010).

La extracción de relación es una tarea de nivel superior, que por lo general depende del buen desempeño de algunas tareas de bajo nivel, tales como tokenización (cf. apartado 2.1.2), separación de frases (cf. apartado 2.1.1) o etiquetado gramatical (ver apartado 2.1.4), y otras tareas de alto nivel, tales como el reconocimiento de entidad (cf. apartado 2.2). La extracción de las relaciones ha llevado la comunidad de minería de textos biomédicos a hacer uso de los textos completos en lugar de limitarse a los resúmenes de las publicaciones, ya que estos últimos son por lo general muy pobres en relaciones.

En cuanto a los métodos utilizados para la extracción de las relaciones biomédicas, (Zhou and He 2008) los clasifican en tres clases: la co-ocurrencia, la coincidencia de patrones y el aprendizaje de máquina. Sin embargo, otros autores (Faro, Giordano et al. 2011) consideran sólo dos enfoques, co-ocurrencias y procesamiento del lenguaje natural. El primer abordaje que ha sido propuesto para la extracción de relación se basó en concurrencias y la coincidencia de patrones. Más tarde, métodos de procesamiento de lenguaje natural han sido utilizados con el fin de tratar relaciones más complejas. Además, el uso de procesamiento de lenguaje natural ha permitido una mejor comprensión del contexto con el fin de tener en cuenta, por ejemplo, la negación y la especulación. Enfoques híbridos que se componen de un o más de esos abordajes también han sido propuestos (Tikk, Thomas et al. 2010).

Co-ocurrencia supone que las entidades que aparecen cerca en el mismo texto, por lo general en la misma frase, están relacionadas entre sí. Se ha utilizado para la extracción de relaciones binarias, tales como genes y enfermedades (Tsuruoka, Tsujii et al. 2008). Sin embargo, este enfoque es incapaz de identificar negaciones o especulaciones, ya que sólo la presencia de las entidades es considerada, pero no el contexto en que ésas aparecen.

El enfoque basado en reglas utiliza reglas manuales o patrones para definir posibles relaciones, por lo general dentro de una oración. Este método puede ser utilizado en conjunto con métodos estadísticos para la estimación de la confianza de una relación.

Aunque el uso de reglas manuales pueden proporcionar buenos resultados, tales como el trabajo de (Kilicoglu and Bergler 2009), un gran esfuerzo es necesario con el fin de construir esos patrones y el sistema resultante no puede ser fácilmente adaptado a otros dominios.

El enfoque de la lingüística computacional analiza la sintaxis y la semántica de un texto con el fin de obtener relaciones entre las entidades predefinidas. El pre-procesamiento del texto incluye por lo general tokenización (cf. apartado 2.1.2),

etiquetas gramaticales (cf. apartado 2.1.4) y análisis sintáctico (cf. apartado 2.1.6). Sin embargo, el análisis de texto biomédico sin restricciones puede ser extremadamente difícil y el rendimiento del análisis tiene una influencia directa sobre el desempeño de esta metodología. Por lo general, este método sólo se utiliza dentro de una frase. Con el fin de detectar relaciones entre frases distintas, un sistema necesitaría llevar en cuenta las anáforas.

El análisis sintáctico superficial (cf. apartado 2.1.6) permite la identificación de las conjunciones de coordinación y la negación y por lo general funcionan bien para la extracción de relaciones sencillas (binarias) (Giuliano, Lavelli et al. 2006). Sin embargo, su rendimiento disminuye considerablemente para las relaciones más complejas y cláusulas relacionadas. Mediante el uso de análisis sintáctico profundo (cf. apartado 2.1.6), más precisión puede ser lograda y relaciones complejas pueden ser identificadas, que por lo general no pueden ser reconocidas por el análisis sintáctico superficial. En el BioNLP Event Extraction Shared Task, los sistemas que han obtenido los mejores resultados utilizaron análisis sintáctico profundo (Kim, Ohta et al. 2009). Otros estudios en este campo también han sugerido que el uso de análisis sintáctico (Miyao, Sagae et al. 2009) mejora el rendimiento de la extracción de las relaciones entre entidades.

El problema de la utilización de un análisis sintáctico profundo es su alta complejidad y esfuerzo computacional. Otros métodos lingüísticos computacionales se han utilizado como en el trabajo de (Yakushiji, Tateisi et al. 2001) y en el sistema Relex (Fundel, Kuffner et al. 2007) para la extracción de una variedad de relaciones. Este enfoque se utiliza generalmente combinado algoritmos de aprendizaje de máquina.

Por último, uno o varios algoritmos de aprendizaje de máquina se pueden utilizar para deducir las relaciones sin la necesidad de definir un conjunto de reglas o gramática. Sin embargo, una colección de documentos anotados con las relaciones y sus entidades correspondientes suele ser necesario. Una variedad de algoritmos de aprendizaje de máquina han sido utilizados en el BioNLP Event Extraction Shared Task, tales como C4.5 (Mora, Farkas et al. 2009), Support Vector Machines (Bjorne, Heimonen et al. 2009) y razonamiento basado en casos (Neves, Carazo et al. 2009), como se propone en esta tesis (cf. apartado 3.4). Una solución similar es el algoritmo basado en memoria implementado por (Morante, Van Asch et al. 2009).

1.3.3 Normalización de menciones

La normalización de entidades biológicas, también conocido como el reconocimiento automático de términos (Ananiadou and Nenadic 2006), incluye no sólo la identificación de las entidades en el texto, sino también la asociación de cada mención a su identificador único en una base de datos específica u ontología.

Muchas son las dificultades en la tarea de normalización de entidades biológica, y en particular para la normalización de genes y proteínas. La variabilidad y la ambigüedad son dos de los problemas más importantes debido a los varios sinónimos que pueden existir para una entidad en particular. Además, estos diversos nombres pueden ser escritos en diferentes formas por distintos autores, no existe una normativa de nomenclatura para la mayoría de los organismos. Por ejemplo, un dado sinónimo puede aparecer en mayúsculas o minúsculas, o con el uso de los espacios o guiones en partes del nombre, como por ejemplo entre las letras y números en su composición. Los nombres también pueden variar debido a errores en la ortografía.

El amplio uso de abreviaturas es también un grave problema ya que un sinónimo puede hacer referencia al nombre completo o a la abreviatura. Además, las letras que componen el acrónimo no siempre se corresponden con las palabras que lo componen. El acrónimo puede incluir alguna letra de más, o todo lo contrario, es posible que falten algunas de las palabras del término. Además, una abreviatura no necesariamente se refiere a las entidades que lo rodean; ésta puede hacer referencia a otros procesos o entidades que no están ni siquiera relacionados con el dominio de la biología molecular.

La ambigüedad es también una cuestión importante en la tarea de normalización de la entidad, ya que un sinónimo particular podría referirse a las diferentes entidades de un mismo organismo o incluso de organismos distintos. La decisión de la especie a la que hace referencia el sinónimo es habitualmente solucionada con el uso de la información de contexto. Por último, los sinónimos pueden coincidir con palabras comunes, como las palabras en inglés, lo que puede terminar siendo detectados por el sistema, cuando no se utiliza una lista de stopwords.

Muchas soluciones han sido propuestas para la normalización de genes y proteínas y la mayoría de ellas comparten la misma secuencia de pasos: (a) la extracción de las menciones del texto; (b) comparación entre la mención y un diccionario pre-procesado de sinónimos, uno para cada uno de los organismos involucrados. Además, un último paso opcional incluye el filtrado de los resultados y/o la realización de una desambiguación entre los candidatos a identificadores, en caso de que más de uno haya sido encontrado para una misma mención.

El primer paso, la extracción de los genes y proteínas se realiza generalmente por el mismo sistema responsable de la normalización (Fundel, Guttler et al. 2005), pero a veces se lleva a cabo por una o más de los sistemas disponibles libremente, tales como Abner (Settles 2005) o Banner (Leaman and González 2008).

El segundo paso, la tarea de normalización, es altamente dependiente del organismo bajo estudio. Por ejemplo, la nomenclatura de los genes y proteínas para la Saccharomyces cerevisiae (levadura) es relativamente simple, mientras que la nomenclatura de la Drosophila melanogaster (mosca de la fruta), a veces coincide con algunas palabras del inglés. Por lo tanto, diferentes organismos pueden requerir diferentes estrategias (Crim, McDonald et al. 2005) o diccionarios específicos (Fundel, Guttler et al 2005; Hanisch, Fundel et al. 2005), dependiendo de la complejidad de su nomenclatura y del grado de ambigüedad de los sinónimos.

La tarea de normalización de genes y proteínas ha recibido mucha atención de la comunidad científica en los últimos años debido a las competiciones de BioCreative (Hirschman, Colosimo et al. 2005; Morgan, Lu et al. 2008). Sistemas independientes, como GNAT (Hakenberg, Plake et al. 2008) y basados en Web, tales como Whatizit (Rebholz-Schuhmann, Arregui et al. 2008) están disponibles para llevar a cabo tareas de normalización.

1.3.4 Medidas de evaluación

Para la evaluación de los resultados para todas las tareas que se examinan en esta tesis, se utilizan los conceptos de precisión, cobertura y F-measure (Shatkay and Feldman 2003). El primer paso para el cálculo de estas medidas es contar el número de resultados correctos e incorrectos que han sido obtenidos. La respuesta correcta será dada por un corpus, que ha sido anotado manualmente por expertos en el dominio, tal como en el trabajo de (Krallinger, Morgan et al. 2008). Por lo tanto, basado en los resultados correctos proporcionados por un corpus de referencia determinado para una determinada tarea, los siguientes valores son calculados: los positivos verdaderos (el número de respuestas correctas realizadas por el sistema), los falsos positivos (el número de respuestas incorrectas realizadas por el sistema) y los falsos negativos (el número de respuestas que están presentes en el corpus pero que no fueron encontrados por el sistema).

Con base en los tres valores anteriores, podemos definir las métricas de precisión, cobertura y F-Measure. Precisión es la fracción de respuestas correctas entre todos los resultados retornados por el sistema. Por lo tanto, es la relación entre los positivos verdaderos y la suma de los positivos verdaderos y positivos falsos, es decir, todos los resultados que han sido devueltos por el sistema. Cobertura es la fracción de respuestas correctas devueltas por el sistema que están de acuerdo con el corpus de referencia. Por lo tanto, es la relación entre los positivos verdaderos y la suma de positivos verdaderos y negativos falsos, es decir, todos los resultados del corpus oficial. F-Measure es la media armónica entre la precisión y la cobertura.

1.4 Resultados y Aportaciones

1.4.1 Razonamiento basado en casos

Razonamiento basado en casos (RBC) (Aamodt and Plaza 1994) es un abordaje perteneciente a la inteligencia artificial y el campo de aprendizaje de máquinas. Consiste en utilizar conocimiento específico a partir de ejemplos del pasado (los casos) para resolver un nuevo problema. Se lleva a cabo mediante la búsqueda de un caso pasado similar de forma a utilizarlo para la solución del nuevo problema. En otras palabras, las nuevas soluciones se infieren (o se recuerdan) mediante la solución de los casos anteriores. Un caso puede ser definido como una situación pasada que ha sido apropiadamente guardada con el fin de ser capaz de ser reutilizada para resolver problemas en el futuro. Un nuevo caso es entonces un nuevo problema a la espera de ser resuelto. RBC se considera como un aprendizaje sostenido ya que nuevos casos resueltos pueden ser retenidos con el fin de ser utilizado para problemas en el futuro. Nuevos casos también se pueden guardar cuando la solución para una determinada situación se ha resuelto con éxito. Además, la razón de la falla puede también ser retenida por el sistema con el fin de evitar que el mismo error vuelva a pasar.

Una de las ventajas de RBC con respecto a otros métodos de aprendizaje de máquina es que, por lo general, no realiza ninguna generalización de la solución. En su lugar, información específica de la situación pasada es utilizada para la solución de los nuevos, que es generalmente más fácil que realizar una generalización. Sin embargo, RBC también puede representar generalizaciones, ya que los casos pueden representan una situación única o una serie de casos similares. Con el fin de ser eficaz, un sistema de RBC debe ser capaz de representar efectivamente a la situación anterior e integrar cada caso en una base de conocimientos, así como ser capaz de recuperar un caso similar en un período de tiempo apropiado.

El ciclo de RBC puede consistir en cuatro pasos, los llamados "cuatro R" (del inglés): recuperar (retrieve), reutilizar (reuse), revisar (revise) y retener (retain). Los casos se guardan en una base de conocimiento y pueden ser recuperados para su reutilización, cuando se necesita una solución para un nuevo caso. Puede pasar que la solución tenga que ser revisada con el fin de ajustarse al nuevo caso. En caso de que una solución sea correcta, el nuevo caso puede ser guardado para uso futuro.

En los sistemas de RBC, una base de conocimientos (o una memoria de casos) necesita ser desarrollado en el fin de permitir la búsqueda y la comparación de los casos. Muchos métodos han sido propuestos para la integración de un nuevo caso

en la memoria. La construcción de una base de la memoria por lo general incluyen las siguientes tareas: la búsqueda de una estructura adecuada para definir el contenido del caso y su organización e indexación de forma a permitir una recuperación, reutilización y retención efectiva.

En RBC, conocimiento del dominio (general) puede ser utilizado para apoyar a los cuatro pasos anteriores, aunque no es obligatorio. Por ejemplo, para el reconocimiento de genes y proteínas, un diccionario de sinónimos para estas entidades podría ser utilizado. En contraste con el conocimiento general, la base de conocimientos donde los casos se guardan representa el conocimiento específico. En las metodologías basadas en RBC propuestas en esta tesis, poco o ningún conocimiento general se ha empleado.

No tenemos conocimiento de que RBC (o métodos similares) hayan sido ampliamente utilizados en el ámbito de minería de texto biomédica, pero algunos trabajos anteriores han sido reportados. RBC ha sido utilizado anteriormente para la clasificación de términos biomédico por el sistema MaSTerClass (Spasic, Ananiadou et al. 2005). Un abordaje basado en memoria (Morante, Van Asch et al. 2009) ha sido propuesto para la extracción de los eventos biológicos, la misma tarea para la cual hemos desarrollado algunas metodologías (cf. apartados 3.3 y 3.4). Sin embargo, RBC ha sido bastante utilizado en otros dominios de la minería de textos (Weber, Ashley et al. 2005).

1.4.2 Metodología general

Nuestra metodología general propone el uso de razonamiento basado en casos para la extracción de entidades biomédicas y sus relaciones. El procedimiento descrito se ha evaluado para la extracción de disparadores de eventos biológicos (cf. apartado 3.3), tarea similar al reconocimiento de entidades, y para la extracción de relaciones, a saber: extracción de los eventos biológicos (cf. apartado 3.4) y asociaciones entre enfermedades y tratamientos (cf. apartado 3.5). La metodología general es común para todas estas tareas, sin embargo, algunas particularidades han sido llevadas a cabo para cada uno de ellas, sobre todo las características que pueden variar dependiendo de los tipos de entidades o las relaciones que se consideran.

La metodología general consiste en los pasos de entrenamiento y test del algoritmo de razonamiento basado en casos. En la etapa de entrenamiento, varios casos son almacenados en una o más bases de los casos. Los datos de entrada (por lo general el resumen de un documento) se representa como un conjunto de casos que se componen de algunas características predefinidas. En la extracción de información, un caso por lo general representa una parte del texto en lugar del documento completo, como usado en la clasificación de texto, por ejemplo. Por lo tanto, un

caso suele corresponder a una ventana de texto de un tamaño determinado, por ejemplo, una palabra y las cinco palabras que vienen antes y después.

Al crear los casos, el texto de un documento es leído y los casos son guardado en la base de los casos, sin repetición del mismo valor para las mismas características. En su lugar, cada caso tiene un atributo que corresponden a su frecuencia en los documentos de entrenamiento. Una o más características pueden ser definidas como desconocidas durante la etapa de test, aquellas cuya solución se dará por los casos extraídos de la base.

Durante la etapa de test, la misma representación de los casos es utilizada para los casos de entrada, es decir, considerando las mismas características usadas en la etapa de entrenamiento, excepto las que están configuradas como desconocidas. El sistema busca en las bases los casos más similares a estos nuevos casos. Más de un caso podría ser devuelto para un determinado problema y la solución final al problema podría obtenerse, por ejemplo, usando un esquema de votación. La solución final viene dada por la asignación a las características desconocidas del valor de las características correspondientes de los casos que han sido seleccionados.

En nuestra metodología, los casos representan un contexto, es decir, una secuencia de palabras consecutivas en una oración. Siendo un caso, un contexto está compuesto por características. Estas características pueden estar relacionadas con las palabras de su composición o con todo el contexto en sí. Por ejemplo, una característica que representa el lema de la palabra por lo general tendría un valor diferente para cada una de las palabras del contexto. Por otro lado, el tipo de la entidad, por ejemplo, si se trata de un gen o no, es una característica relacionada con todo el contexto. Los límites que fijan los límites del contexto, es decir, el conjunto de palabras, pueden ser definidos de dos maneras en nuestra metodología, a saber: como una ventana de palabras de una longitud predefinida (limitado por algunas palabras predefinidas de inicio y fin) o como una estructura de tamaño variable en función de algunas entidades.

En nuestra metodología, los contextos son generados automáticamente con base en las entidades que han sido previamente identificadas y de la tarea que se analiza. Por ejemplo, para el corpus BioText (cf. apartado B.5), sólo dos entidades están implicadas, una enfermedad y un tratamiento. Sin embargo, los eventos biológicos (ver apartado B.4) son mucho más complejos ya que pueden tener muchos tipos de argumentos (entidades u otros eventos).

Por lo tanto, el primer paso en la generación de los candidatos para los distintos contextos es identificar las entidades en el texto y separarlas en "bolsas de entidades" para cada frase. Dadas las bolsas de las entidades, los contextos son

generados automáticamente mediante la combinación de una o más entidades de cada una de las bolsas, de acuerdo con el tipo de relación bajo consideración. La longitud de un contexto depende de la representación del contexto utilizado (cf. apartado 3.2.1). Después de la generación de los contextos, éstos son convertidos en casos, que luego se insertan en la base de los casos. Este procedimiento es similar para las etapas de entrenamiento y test.

Durante la etapa de pruebas, los casos son recuperados de la base de casos con el fin de ser utilizado como solución. Como entrada, los casos que representan los contextos del texto de test son utilizados, de acuerdo con la representación elegida (cf. apartado 3.2.1). Aquí la búsqueda de un caso se lleva a cabo de dos maneras, mediante una búsqueda binaria por casos que coinciden exactamente con tantas características como sea posible con el caso de entrada, o mediante la realización de un alineamiento global entre algunos de los casos de la base con el caso de entrada. Esta metodología se ha utilizado anteriormente como parte de un algoritmo de RBC para la clasificación biomédica en el sistema MasterClass (Spasic, Ananiadou et al. 2005). Más de un caso podría ser devuelto para un dado caso de entrada. La solución final se obtiene con base en un esquema de votación.

Esta sección presentamos la metodología para la extracción de los eventos biológicos con base en el corpus de BioNLP'09 Event Extraction Shared Task (cf. apartado B.4). Este corpus contiene anotaciones para nueve tipos de eventos biológicos: localización, binding, expresión génica, transcripción, catabolismo proteico, fosforilación, regulación, regulación positiva y regulación negativa. En este corpus, las proteínas han sido proporcionadas y no necesitan ser previamente extraídas. Sin embargo, es necesaria efectuar la extracción de los disparadores de los eventos, una tarea de reconocimiento de entidades. Proponemos dos métodos para la extracción de disparadores de eventos, el que participó en la competición de BioNLP'09 Event Extraction Shared Task (Neves, Carazo et al. 2009), que se basa principalmente en análisis sintáctica superficial (cf. apartado 2.1.6), y una mejora de esta metodología, que utiliza algunas características del análisis sintáctico profundo (cf. apartado 2.1.6).

El enfoque utilizado para la extracción de los factores desencadenantes de eventos se basa en la metodología general. Algunos de los procedimientos específicos que se utilizan sólo para esta tarea son descritos aquí. Las características que componen un caso, tanto durante el entrenamiento y el test son las siguientes: la propia palabra; la palabra en minúsculas; la raíz de la palabra (Porter stemmer); la forma de la palabra; la etiqueta gramatical (GENIA tagger); la etiqueta de la frase (chunker - GENIA tagger); el tipo de entidad (GENIA tagger); el tipo del entidad (Protein, Entity, GeneExpression, etc.); el tipo de evento (GeneExpression, Localization, etc.); y la parte del evento (Location, Theme, Cause, etc.).

Por lo general, uno caso es creado para cada palabra de los documentos de entrenamiento. Los casos son una representación de una ventana de contexto (-1,0), es decir, para cada palabra, se procesan las características de la propia palabra y de la anterior, exclusivamente.

La estrategia de búsqueda de casos utilizada para la similitud se basa en la coincidencia exacta de las características (cf. apartado 3.2.3.1), es decir, el sistema intenta encontrar un caso con el mayor número de características que tenga exactamente el mismo valor de las respectivas características del caso de entrada. La raíz de la palabra es la única característica obligatoria que tiene que ser cumplida. El mejor caso de los casos entre aquellos recuperado por el sistema es él que tenga la frecuencia más alta. El valor de las características desconocidas se dará por los valores de las características respectivas del caso seleccionado. Si ningún caso se recupera, la palabra no es considerada como parte de un evento biológico.

Como la competición de BioNLP'09 Event Extraction Shared Task no incluía una evaluación en separado para los disparadores de eventos y demás entidades, hemos llevado a cabo nuestra propia evaluación del conjunto de datos de desarrollo con el fin de comprobar el desempeño de esta tarea. Nuestro sistema está más orientado a la obtención una alta cobertura, ya que una entidad no reconocida en este paso no será considerada para la extracción de los eventos biológicos (cf. apartado 3.4). Por otro lado, falsos disparadores extraídos en este paso pueden ser clasificados como negativos durante el siguiente paso, si no se encuentran argumentos relacionados a los disparadores. La cobertura que hemos obtenido ha variado según el tipo de disparador, y va desde casi 95% para la fosforilación hasta 55% para el reconocimiento de localizaciones.

En este segundo enfoque para el reconocimiento de los disparadores y demás entidades, un caso también representa una ventana de palabras (ver apartado 3.2.1.1), pero esta vez de tamaño [-1,1], es decir, incluye tanto a la palabra anterior como la siguiente. Esta configuración se decidió después de llevar a cabo algunos experimentos variando el tamaño de la ventana. En este segundo abordaje, también hacemos predicciones para el modificador del evento, es decir, si hay negación o especulación relacionada con el evento. Esta predicción se realiza junto con el reconocimiento del disparador del evento.

Las características utilizadas para representar cada elemento (palabra) de la ventana son las siguientes: lema de la palabra (Dragón Toolkit (Zhou, Zhang et al 2007)); etiqueta gramatical (Stanford parser (Klein y Manning 2003)); distancia a la proteína más cercana (número de palabras, en múltiplos de cinco); dirección de la proteína más cercana (derecha o izquierda); distancia en términos de las etiquetas de dependencia a la proteína más cercana, en múltiplos de dos (de

Marneffe, MacCartney et al 2006); tipo de la entidad junto con la etiqueta BIEWO y el modificador ("ninguno", "especulación" o "negación"). La estrategia de búsqueda utilizada fue la coincidencia exacta de las características (véase apartado 3.2.3.1).

Una vez más, hemos llevado a cabo nuestra propia evaluación del conjunto de datos de desarrollo con el fin de comprobar el desempeño del sistema. Los resultados obtenidos con este abordaje han tenido en general una cobertura inferior a la metodología anterior, pero un mejor F-Measure.

1.4.3 Extracción de eventos biológicos

Proponemos dos metodologías para la extracción de los eventos, es decir, la relación entre los disparadores de eventos (extraídos anteriormente en los apartados 3.3.1 y 3.3.2) y sus argumentos. Los argumentos pueden ser una proteína o una localización celular, también extraído anteriormente en los apartados 3.3.1 y 3.3.2. El primer método está basado en reglas manuales y participó en el BioNLP'09 Event Extraction Shared Task (Neves, Carazo et al. 2009) (véase apartado 3.4.1), mientras que el segundo utiliza la metodología general descrita en el apartado 3.2 para el razonamiento basado en casos (cf. apartado 3.4. 2).

Para la primera aproximación, los disparadores de eventos han sido extraídos utilizando la metodología basada en análisis sintáctico superficial (cf. apartado 3.3.1). Nuestro enfoque busca extraer los argumentos de forma incremental, usando los disparadores como punto de partida. El orden en que los argumentos son extraídos del texto es el siguiente: “theme”, “theme2”, “cause”, “location” y “site”. Las reglas se basan en los valores asignados para las tres características desconocidas del apartado 3.3.1: tipo de entidad, tipo de evento y parte del evento.

Los candidatos para el argumento "theme" son las proteínas anotadas, así como los eventos mismos en el caso de la regulación. La estrategia de búsqueda se inicia desde el disparador de eventos, una palabra de cada lado hasta que un candidato sea encontrado. El sistema se detiene si encuentra el final de la frase o si alcanza 20 palabras en cada dirección. Con respecto al segundo tema (“theme2”), utilizamos una estrategia de búsqueda similar, excepto que ahora el sistema lee un máximo de 10 palabras en cada dirección, a partir del primer argumento “theme” previamente extraído. Lo mismo ocurre con el argumento “cause”, lo cual la búsqueda se detiene hasta 30 palabras en cada dirección desde el disparador de eventos. Por último, para la localización celular, el límite es de 20 palabras en cada dirección desde el argumento “theme”.

La evaluación de esta metodología ha sido llevada a cabo con durante la competición del BioNLP'09 Event Extraction Shared Task y los resultados varían muchos según el tipo de evento. El mejor resuñtado alto ha sido para el

catabolismo proteico (65% de F-Measure) y el más bajo el de regulación (6% de F-measure).

También hemos utilizado el razonamiento basado en casos para la extracción de las relaciones los argumentos de los eventos biológicos. El caso se representa como un contexto (cf. apartado 3.2.1.2), cuya longitud se define automáticamente a partir de algunas palabras predefinidas de la frase, es decir, las entidades que podrían estar involucradas en el evento.

Dado que la extracción de evento es una tarea compleja, muchos cambios son necesarios en el algoritmo general (cf. apartado 3.2) con el fin de adaptarlo a las particularidades del dominio. La construcción de las bolsas de las entidades (cf. apartado 3.2.2) es bastante simple y lleva en cuenta el tipo de entidad de la palabra: las proteínas van a la bolsa de "proteínas", localizaciones a la de "Entidades" y los disparadores de eventos a la de "Eventos".

La generación automática de los contextos es una de las tareas más complejas, debido a que el corpus incluye muchos tipos de eventos (regulación, localización, expresión de genes, etc.) que pueden estar compuestos de distintos argumentos de distintos tipos. Afortunadamente, el problema se limita a la cantidad de entidades dentro de cada bolsa. El número de candidatos a eventos para un cierto disparador puede aumentar rápidamente dependiendo de la cantidad de entidades presentes en las bolsas.

Debido a limitaciones de tiempo y con el fin de no generar contextos extremadamente largos, que no serían muy útil durante la fase de test, hemos limitado el tamaño del contexto a 20 palabras. Los contextos son siempre delimitados por las entidades que se encuentran más a la derecha y más a la izquierda en su composición. Los casos que representan un contexto de eventos se componen de las siguientes características: indicativo de si hay una relación (evento), un texto representado el evento y sus argumentos (e.g., “AtLoc:Entity,Trigger:Localization,Theme:Protein”), y respeto a cada palabra del contexto: la etiqueta gramatical, el tipo de entidad y el rol de la misma en el evento (“theme”, “cause”, etc.).

La evaluación de la metodología se llevó a cabo con los conjuntos de desarrollo y test (cf. apartado B.4). Los resultados dependen de cada tipo de evento y van desde aproximadamente 15% de F-measure para el evento de regulación y 61% de F-measure para la expresión génica.

1.4.4 Extracción de relaciones entre enfermedades y tratamientos

En esta sección, aplicamos una vez más nuestra metodología general del razonamiento basado en casos (cf. apartado 3.2) para la extracción de las

relaciones biomédicas. En esta ocasión, hemos decidido probar con un problema más sencillo, la extracción de las relaciones entre enfermedades y tratamientos con el corpus de BioText (cf. apartado B.5).

El corpus BioText no es tan complejo como el de eventos biológicos. Aquí, las entidades que participan en las relaciones son fornecidas. Además, sólo hay una relación por frase y esta relación siempre está compuesta de una enfermedad y un tratamiento. Además, no hay necesidad de identificar la relación dentro de la frase, ya que el corpus está anotado a nivel de frase. Sin embargo, las relaciones deben ser clasificadas en las siguientes clases: "PREVENT" (prevención), "SIDE_EFF" (efectos secundarios), "VAGUE" (incierto), "TREAT_ FOR_DIS" (hay tratamiento de la enfermedad) y "TREAT_NO_FOR_DIS" (no hay tratamiento de la enfermedad).

Sólo algunos pocos cambios en el algoritmo general han sido necesarios para las particularidades del dominio. Sólo tuvimos que decidir qué características son más apropiadas para representar el contexto e implementar la generación de esos contextos, que consta de una sola instancia de cada tipo de entidad, una enfermedad y un tratamiento. En cuanto a las características, a nivel del contexto, se utiliza el tipo de relación y un texto con el orden de las entidades ("DIS,TREAT" y "TREAT,DIS"). A nivel de palabra, se utiliza la etiqueta gramatical, el tipo de entidad, el rol de la entidad y el lema.

Para la evaluación de la metodología, se ha realizado una validación cruzada en cuatro iteraciones con las 964 frases que componen el corpus En cada iteración, el 75% de las sentencias se utiliza para entrenamiento y el 25% para test. Los resultados varían mucho según el tipo de relación y va desde un F-measure nulo para los tipos SIDE_EFF y TREAT_NO_FOR_DIS debido a los pocos ejemplos disponibles para entrenamiento, a 91% de F-measure para el tipo TREAT_FOR_DIS.

1.4.5 Reconocimiento de genes y proteínas

Para la extracción de los genes y las proteínas de la literatura científica, proponemos el uso de razonamiento basado en casos (Aamodt y Plaza, 1994). En este sentido, nuestro enfoque es diferente de la metodología general que se propone en la sección 3.2. Para la etapa de entrenamiento y test, se han utilizado los corpora de BioCreative Gene Mention task (cf. apartado B.1), que consiste en 15.000 y 5.000 frases, respectivamente. Durante este primer paso, el conjunto de datos de entrenamiento se dividió en 10 subconjuntos con el fin de realizar una validación cruzada.

Las palabras fueron extraídas de los documentos y son utilizadas para construir las dos bases de casos, una de los casos conocidos y otra para los casos desconocidos,

como se propone para el problema de etiquetado gramatical en (Daelemans, Zavrel et al., 1996). El clasificador también ha sido entrenado con corpora adicional con el fin de ser capaz de extraer menciones de diferentes organismos. Estos corpora adicionales pertenecen a los conjuntos de datos para normalización de genes del BioCreative task 1B (cf. apartado B.1) para la levadura, el ratón y la mosca.

Los casos conocidos son utilizados por el sistema para clasificar las palabras que han aparecido en los documentos de entrenamiento. Los atributos utilizados para representar un caso conocido son los siguientes: la propia palabra; la categoría de la palabra (si es una mención de gen o no); y la categoría de la palabra anterior (si es una mención de gen o no).

La base de casos desconocidos se utiliza para clasificar las palabras que no estaban presentes en los documentos de entrenamiento. Los casos desconocidos se construyen sobre los mismos datos de entrenamiento utilizados para los casos conocidos. En lugar de guardar la palabra en sí, se usa la forma de la misma con el fin de permitir que el sistema pueda clasificar palabras desconocidas. Los atributos que han sido utilizados para representar los casos desconocidos son los siguientes: la forma de la palabra, la categoría de la palabra (si es una de mención gen o no); y la categoría de la palabra anterior (si es una mención de gen o no).

En la construcción de los casos, los documentos de entrenamiento se leen dos veces, de izquierda a derecha, y de derecha a izquierda. Esto se hace de forma a permitir una mayor variedad de casos debido al hecho de que la decisión de clasificar una palabra puede estar influenciada por la palabra anterior o siguiente.

El procedimiento de búsqueda se divide en dos partes, una para los casos conocidos y otro para los casos desconocidos. En esta estrategia de búsqueda, se da prioridad a los casos conocidos frente a los desconocidos. Después de la búsqueda del mejor caso para cada palabra, un post-procesamiento es ejecutado con el fin de comprobar los límites de la mención, así como abreviaturas y nombres completos correspondientes.

Una versión inicial de este metodología (Neves 2007) ha participado en el BioCreative 2 Gene Mention task (Smith, Tanabe et al. 2008). Una de las diferencias entre estos dos enfoques es que el que aquí se describe lee los documentos en ambas direcciones. Además, algunas mejoras se han incluido con la metodología, tales como cambios en la forma de los casos desconocidos, incluyendo la consideración de sufijos y prefijos, así como las etapas de procesamiento posteriores a fin de tener en cuenta los límites de la mención y abreviaturas.

Hemos evaluado nuestro sistema con los 5.000 documentos que componen el BioCreative 2 Gene Mention corpus (véase B.1). Los resultados varían según el conjunto de entrenamiento utilizado, si solo el disponible para el BioCreative 2 Gene Mention task o uno o más de los corpora disponibles para levadura, ratón y mosca. El mejor F-measure (casi 70%) ha sido obtenido solo con los datos del BioCreative 2 Gene Mention corpus mientras que el peor resultado (37% de F-measure) cuando se utiliza además los tres otros corpora específicos. Sin embargo, los resultados presentados en la sección 4.2, para la normalización de genes y proteínas, muestran que en algunos casos, dependiendo del organismo en consideración, un clasificador entrenado con documentos específicos del organismo puede mejorar el F-measure para esta tarea.

1.4.6 Normalización de menciones de genes y proteínas

La tarea de normalización de menciones de genes y proteína es el problema de asociar un identificador en un diccionario de sinónimos (que puede ser específico de un dado organismo) a una potencial mención de gen o proteína previamente reconocida en el texto. Proponemos tres metodologías para esta tarea: coincidencia exacta a partir de un listado de sinónimos previamente editado (cf. apartado 4.2), coincidencia aproximada basada en un trie y alineamiento global (cf. apartado 4.3) y coincidencia aproximada basado en aprendizaje de máquina (cf. apartado 4.4).

Para todas las metodologías que aquí se propone, el diccionario de sinónimos es utilizado para la construcción del sistema de normalización se basa en los proporcionados por las competiciones de BioCreative, que contiene 14.995 sinónimos para la levadura, los 130.208 sinónimos para el ratón, 116.744 sinónimos para la mosca (cf. apartado B.2) y 203.077 sinónimos para el humano (cf. apartado B.3).

La concordancia exacta consiste en verificar la coincidencia entre una mención y los sinónimos en los diccionarios. Sin embargo, la correspondencia no es realizada con la mención y sinónimos originales, pero con variantes de ambos. La mención y el diccionario de sinónimos son previamente pre-procesados mediante algunas operaciones de edición. Estas operaciones se llevan a cabo sólo una vez para el diccionario de sinónimos durante el desarrollo del sistema. Sin embargo, tienen que ser realizadas para cada mención durante el procedimiento de normalización. Las operaciones de edición se llevan a cabo igualmente para cualquier organismo.

En primer lugar, las palabras son convertidas a minúsculas y sus partes separadas de acuerdo con algunos límites, como símbolos, puntuación, letras griegas y números. Estas partes son a continuación ordenadas alfabéticamente, como se propone en (Liu, Wu et al. 2004), con el fin de evitar que no haya una coincidencia debido al orden diferente de los mismos símbolos. El sistema también realiza un filtrado de la mención (o sinónimo) con el fin de eliminar partes del mismo que

coincidan con términos biomédicos del listado de BioThesaurus, stopwords o nombres de organismos de la base de datos del NCBI Entrez Tazonomy. La limpieza de los términos biomédicos del listado de BioThesaurus se lleva a cabo gradualmente según la frecuencia del término en el listado. Este procedimiento genera muchas variaciones de la mención (o sinónimo) original. Con este procedimiento, se aumenta la posibilidad de encontrar una coincidencia exacta, sin necesidad de proporcionar información específica de un organismo.

La evaluación es realizada con los conjuntos de datos del BioCreative task 1B para la levadura, el ratón y la mosca (cf. apartado B.2) y del BioCreative II Gene Normalization task (cf. apartado B.3) para el humano. Los resultados varían según el organismo y va desde 63% de F-measure para la mosca a casi 89% de F-measure para la levadura. También hemos realizado experimentos con distintos sistemas para la extracción de las menciones: CBR-Tagger (cf. apartado 3.6), ABNER (cf. apartado C.5) y BANNER (cf. apartado C.6).

El segundo método que proponemos para la normalización de genes y proteínas utiliza en un primer intento una correspondencia exacta entre las menciones a los sinónimos del diccionario, la estrategia de coincidencia más rápida posible. Para esta coincidencia exacta se utiliza el diccionario original (cf. 4.1) previamente convertido a minúsculas. En los casos en que no ocurra una coincidencia exacta, llevamos a cabo una coincidencia aproximada con base en un alineamiento global de las menciones y los sinónimos del diccionario (Neves, Chagoyen et al. 2008). Los diccionarios de sinónimos considerados son los descritos en la sección 4.1. La única operación de edición llevada a cabo ha sido convertir los sinónimos a minúsculas.

Por lo general, una correspondencia aproximada requiere que cada mención sea comparada con cada sinónimo del diccionario, lo que puede consumir mucho tiempo. De forma a evitar este problema y con el fin de mejorar el rendimiento del sistema, los sinónimos han sido guardados en una estructura de trie (un árbol de recuperación) (Shang y Merrettal 1996). En un trie, cada palabra está representada por los nodos de un solo carácter según una estructura de árbol. Palabras con el mismo prefijo se encuentran en las mismas ramas del árbol.

La ventaja de usar un trie es que no hay necesidad de realizar repetidas alineaciones cuando se compara una mención con sinónimos con los que comparten el mismo prefijo. Además, la búsqueda a través de una determinada rama puede ser abortada si el coste mínimo de la alineación en esta rama es mayor que un umbral predefinido. El resultado de esta estrategia es una reducción en el tiempo de procesamiento sin sacrificar la calidad de la comparación entre la mención y los sinónimos.

Nuestra coincidencia aproximada realiza un alineamiento global basada en costes predefinidos, como se sugiere en (Tsuruoka and Tsujii 2003) para el problema de reconocimiento de genes. Para la alineación global se usa la distancia de edición entre dos cadenas (una mención y un sinónimo). Esto se realiza utilizando un algoritmo de programación dinámica y puede ser mejor definido como el número mínimo de operaciones (exclusión, inserciones y sustituciones) necesarios para llevar a cabo a nivel de carácter con el fin de transformar una cadena en otra.

Los costes iniciales para la inclusión, supresión y sustitución de caracteres fueron los propuestos en (Tsuruoka y Tsujii 2003). Éstos fueron posteriormente adaptados de acuerdo con experimentos llevados a cabo con la levadura, ratón, mosca y humano con los corpora de BioCreative (cf. apartados B.2 y B.3) durante el desarrollo del método.

La evaluación es realizada con los conjuntos de datos del BioCreative task 1B para la levadura, el ratón y la mosca (cf. apartado B.2) y del BioCreative II Gene Normalization task (cf. apartado B.3). Los resultados obtenidos con esta metodología varían según el organismo y van desde casi 50% de F-measure para el ratón y 91% de F-measure para la levadura.

Para la tercera estrategia, proponemos un clasificador binario basado en tres algoritmos de aprendizaje de máquina: Support Vector Machines (Joachims 1998), Random Forests y regresión logística. Este procedimiento sólo se lleva a cabo si la coincidencia exacta no retorna resultados.

Utilizamos la herramienta Weka (Witten y Frank 2005) para entrenamiento y test de los tres algoritmos de aprendizaje automático. Para la etapa de entrenamiento, las características representan la comparación entre un par de sinónimos y la categoría, es decir, si se trata de una coincidencia o no. Por otro lado, en la etapa de test, las características representan una comparación entre una dada mención y los sinónimos del diccionario.

El entrenamiento es un procedimiento en tres pasos. Se utiliza la metodología propuesta en (Tsuruoka, McNaught et al. 2007). Los atributos de los ejemplos de entrenamiento se obtienen mediante una comparación de dos sinónimos del diccionario de acuerdo con algunas características predefinidas. Para cada organismo, la comparación de dos sinónimos de un gen determinado constituirá los ejemplos positivos, mientras que la comparación de dos sinónimos de genes distintos del mismo organismo representa los ejemplos negativos.

Los pasos para la construcción de los datos de entrenamiento incluyen la extracción de características de los sinónimos, seguido por la selección de los pares de sinónimos a ser comparados. En el primer paso, las características que

representan un sinónimo son generadas para todos los sinónimos de los diccionarios. Las características son las siguientes: prefijo (tres primeras letras del sinónimo), sufijo (tres últimas letras del sinónimo), número que forma parte del sinónimo, letra griega que forma parte del sinónimo, bigrams y trigrams (cf. apartado 2.1.3) del sinónimo, y forma del sinónimo.

El segundo paso es la selección de un conjunto de pares de sinónimos para ser comparados, que se componen los ejemplos positivos y negativos utilizados para entrenar los algoritmos de aprendizaje de máquina. Esta es una etapa muy lenta ya que los pares de sinónimos fueron seleccionados de forma que compartan una cierta similitud. Además, intentamos tener un conjunto de datos de entrenamiento equilibrado, es decir, con el mismo número de ejemplos positivos y negativos, y así evitar el exceso de una de las clases.

Las características que componen los ejemplos de entrenamiento en este paso representan una comparación entre un par de sinónimos. Por lo tanto, se obtienen mediante la comparación de las características mostradas anteriormente para los sinónimos en si. Las características son las siguientes: indicativo de si los prefijos son iguales, indicativo de sufijos son iguales, indicativo de si el número es igual, indicativo de la letra griega es igual, similitud de bigrams o trigrams, similitud entre palabras (Levenstein, Jaro-Winkler, Monge-Elkan, Smith-Waterman and SoftTFIDF), y la diferencia entre las formas.

Varios son los parámetros que pueden ser configurados para esta metodología y varios experimentos han sido efectuados de forma decidir los mejores valores para los cuatro organismos (levadura, ratón, mosca y humano). La evaluación es realizada con los conjuntos de datos del BioCreative task 1B para la levadura, el ratón y la mosca (cf. apartado B.2) y del BioCreative II Gene Normalization task (cf. apartado B.3) para el humano. Los resultados obtenidos con esta metodología varían según el organismo y van desde casi 43% de F-measure para el humano a 82% de F-measure para la levadura.

Cuando más de un identificador es encontrado para una mención, un procedimiento de desambiguación es necesario de forma a decidir cual de ellos es más probable de ser correcto. Proponemos una comparación entre el resumen del artículo y un documento representante de cada uno de los genes y proteínas (gen-documento). El gen-documento es construido por la compilación de información extraída de diversas bases de datos, tales como SGD (Cherry, Adler et al. 1998) (cf. apartado A.5) para la levadura, MGI (Eppig, Bult et al. 2005) (cf. apartado A.6) para el ratón, FlyBase (Gelbart, Crosby et al. 1997) (cf. apartado A.7) para la mosca y Entrez Gene (Maglott, Ostell et al. 2007) (cf. aparatdo A.3) para el humano. Los campos recogidos para la construcción de los gen-documentos fueron símbolos, sinónimos, descripciones, resúmenes, productos, fenotipos, relaciones,

interacciones, términos de Gene Ontology (Ashburner, Ball et al. 2000) (véase A.4).

El texto contenido en estos campos es separado por palabras y éstas son guardadas en una bolsa de palabras. Un modelo de espacio vectorial es construido para cada documento y está compuesto por todas sus palabras, excepto los números cardinales y ordinales, unidades de medidas pre-definidas y palabras que coincidan con un listado de stopwords (cf. apartado E.1). De las palabras resultantes, son extraídas su raíz con el Porter stemmer (cf. apartado C.2) y sus pesos ponderados en el documento de acuerdo con el TF-IDF (Salton and Buckley 1988). Este procedimiento se realiza para el cada gen-documento de los candidatos, así como para el artículo en consideración.

Tres métodos de desambiguación pueden ser elegidos. El primero usa la similitud del coseno (Shatkay and Feldman 2003) entre el artículo y los gen-documentos, mientras que el segundo tiene en cuenta el número de palabras comunes entre los dos textos. En el primer caso, el gen-documento con la mayor similitud de coseno es elegido como el identificador correcto para la mención. En el segundo caso, el gen-documento con mayor número de palabras común es elegido como la mejor solución. La tercera metodología se basa en ambas métricas.

Experimentos han sido llevados a cabo y se han tomado en consideración la selección de solo el mejor candidato (desambiguación individual) o de varios según un determinado umbral (desambiguación múltiple).

1.5 Conclusiones y Trabajos Futuros

En este trabajo se han propuesto nuevas metodologías para algunos problemas de minería de texto biomédica, y más específicamente, para la extracción de entidades, la extracción de las relaciones biomédicas y la normalización de menciones. Para las dos primeras tareas, se ha utilizado el abordaje de razonamiento basado en casos. En esta sección empezaremos por la discusión del desempeño de este método para las tareas de minería de texto que son analizadas en esta tesis.

Por lo general, las metodologías propuestas han utilizado poco conocimiento específico del dominio, con la excepción de la desambiguación de genes y proteínas (cf. apartado 4.5), en la que se han utilizado información extraída de bases de datos específicas del genoma de cada organismo. La decisión de proponer metodologías que utilizan poco conocimiento del dominio se ha debido principalmente a dos razones: la decisión intencional de la construcción de un sistema lo más general posible, sin la necesidad de expertos biomédicos, y la inexistencia de los mismos en nuestro equipo, y por lo tanto, la imposibilidad de adquirir dichos conocimientos. El precio a pagar por esta generalidad es, como se

esperaba, el sacrificio del rendimiento, a pesar de que nuestros experimentos demuestran que los métodos presentan resultados satisfactorios en las tareas para las que fueron diseñados, mientras que también podrían mejorar con el conocimiento del dominio.

En cuanto al ciclo del razonamiento basado en casos (cf. apartado 3.1), nuestras metodologías sólo consideran los pasos "recuperar" y "reutilizar", cuando los casos son recuperados de la base de los casos y reutilizados como solución para un nuevo problema, respectivamente. La decisión de no tener en cuenta el paso "revisar" se debió a la imposibilidad de obtener una retroalimentación por parte de los usuarios de los sistemas y, por consiguiente, la incapacidad de salvar el caso revisado para su uso futuro. Dicha revisión sólo podría ser realizada en un conjunto de datos de prueba, pero tal conjunto de datos sólo debe utilizarse para la evaluación y su utilización para la revisión y almacenaje de un nuevo caso no estaría de acuerdo con los principios del aprendizaje de máquina.

Para la extracción de genes y proteínas, un paso necesario en muchos procedimientos de minería de texto biomédica, hemos propuesto el abordaje des razonamiento basado en casos. Los resultados muestran la idoneidad de este enfoque para la tarea. A pesar de que CBR-Tagger no produce los mejores resultados cuando considerado de forma aislada, cuando combinado con otros sistemas (e.g., ABNER o BANNER), nuestros experimentos demostraron que hay una mejora en los resultados para la tarea de normalización de genes y proteínas (cf. apartado 4.2). Aunque los resultados presentados para la extracción de la mención de genes y proteínas parecen indicar que entrenar el sistema con documentos específicos de algunos organismos podría resultar en un peor desempeño del sistema, los resultados presentados para la normalización de los genes y proteínas para la mosca (cf. apartado 4.3) muestran claramente que no es el caso. Consideramos que el reconocimiento de genes y proteínas es un paso anterior para el problema de la normalización y la mejora de éste es el objetivo principal de un sistema de reconocimiento de estas entidades. Además, los documentos específicos de cada organismo que han sido utilizados para el entrenamiento del sistema han sido creado con base en la correspondencia exacta de las menciones de un diccionario de sinónimos, y ningún conocimiento adicional relacionado con los organismos ha sido añadido al sistema, lo que es una ventaja en aquellos casos en que esta información específica no está disponible. El entrenamiento del sistema con estos documentos puede también ayudar a otros organismos distintos, como ha sido el caso del humano.

Como limitaciones de este abordaje, otras características podrían haber sido utilizado, tal como algunas de las descritas en el apartado 2.2.1, así como una ventana más ancha de palabras. Estas limitaciones fueron por lo general debido a limitaciones de tiempo, ya que el reconocimiento de genes y proteínas ha sido la

primera metodología que se ha desarrollado como parte de esta tesis, en principio para la competición BioCreative 2 Gene Mention task (Smith, Tanabe et al. 2008). El razonamiento basado el casos fue entonces mejorada de forma a poder ser usado para la extracción de eventos biomédicos (cf. apartado 3.4). En esta implementación mejorada, los pasos de retención y recuperación de un caso de la base de datos de MySQL es mucho más eficaz con el uso de índices y tabla caché que recuerda consultas ya realizadas anteriormente.

En cuanto a la normalización de los genes y las proteínas (cf. capítulo 5), nuestras metodologías no ha podido alcanzar los niveles de otros sistemas existentes. Sin embargo, no tenemos conocimiento de ninguna otra herramienta disponible gratuitamente que permita su integración y el entrenamiento con nuevos organismos para esta tarea. Este es un punto fuerte en nuestro trabajo, ya que posibilita que mejoras sean implementadas en nuestro sistema. Una vez más, se ha obtenido un F-measure satisfactorio sin necesidad de realizar cambios en los algoritmos de forma a que se adaptara a algún organismo en particular. Además, nuestro sistema ha sido diseñado con poca dependencia de diccionarios personalizados o documentos anotados, que por lo general no están disponibles públicamente. Al comparar nuestros resultados con los reportados en las dos ediciones de BioCreative (Hirschman, Colosimo et al. 2005; Morgan, Lu et al. 2008), hemos encontrado que las que han obtenido mejor F-measure que los nuestros han hecho uso de procedimientos específicos para cada organismo, ya sea por un diccionario creado manualmente o de estrategias específicas de adaptación.

Por lo tanto, podemos afirmar que nuestro sistema de normalización requiere mucho menos dependencia de conocimiento específico del organismo ya que utiliza sólo información que está disponible libremente para la comunidad científica (e.g., bases de datos públicas), y ningún conocimiento específico inferido por expertos, lo que pasa con la gran mayoría de los otros sistemas. La mayoría de los métodos y herramientas que han obtenido un buen desempeño en la tarea de normalización de genes han utilizado información específica para el ajuste del sistema e incluso a reglas manuales específicas. Aunque este abordaje produce buenos resultados para ciertos organismos, el sistema no puede extenderse a nuevos organismos sin un conjunto de reglas similares deducidas expertos. Por lo tanto, la reproducción de los métodos existentes con nuevos organismos toma mucho tiempo y es muchas veces imposible de realizarse. En nuestro sistema, sólo utilizamos la información disponible públicamente para cada organismo. Es cierto que no podemos excluir algo de la información específica para cada organismo, tal como el diccionario de sinónimos o anotaciones de genes y proteínas, que son necesarios para las tareas de normalización y desambiguación, respectivamente. Sin embargo, esta información puede ser obtenida a partir de bases de datos públicas y no ha habido una adaptación específica para cada organismo de forma a obtener resultados satisfactorios.

En el caso de que un nuevo organismo vaya a ser introducido en el sistema, un diccionario de sinónimos e información relacionada con sus genes y proteínas, como la descripción, fenotipo, y términos de Gene Ontology asociado, es el único conocimiento necesario a ser utilizado. Toda esta información suele ser fácilmente obtenida en bases de datos específicas de cada organismo. El cuello de botella aquí es la necesidad de documentos anotados para la evaluación de los resultados. Esa es la razón principal por la que no hemos podido ampliar nuestro sistema con otros organismos además de aquellos cuyos corpora están disponibles en las competiciones de BioCreative. La disponibilidad de documentos pertinentes relacionados a los genes es, en la actualidad, una limitación para los métodos automáticos de minería de texto, y en particular para aquellos métodos que requieren una colección de referencias bibliográficas correspondientes a los genes y proteínas y que no disponen de anotaciones manuales y referencias bibliográfícas asociadas.

Hemos analizado los errores para la normalización de genes y proteínas a partir del conjunto de documentos de desarrollo, ya que ningún análisis ha sido realizado con el conjunto de documentos de test. Algunos de los errores de falsos negativos se debieron a menciones que no pudieron ser extraídas por el sistema, incluso cuando se utiliza una mezcla de tres de ellos. Además, un alto número de falsos negativos se deben a una mal desambiguación, con la consecuente generación de muchos otros falsos positivos. Los resultados previstos con este enfoque parecen muy prometedores y sin duda tienen más margen de mejora, en particular con en el procedimiento de desambiguación. Este procedimiento no estaba originalmente en el foco principal de este estudio, aunque los resultados indican claramente que más esfuerzos deben ser dedicados a ellos ya que la mejora de la normalización depende en gran medida del desempeño de la desambiguación.

La aplicación de nuestra metodología de normalización de genes y proteínas a problemas reales de la minería de datos requiere un manejo con más de un organismo al mismo tiempo. La implementación de esta funcionalidad en el sistema es factible como trabajo futuro y sólo requiere la realización del procedimiento de coincidencia con los diccionarios de sinónimos específicos para cada organismo en consideración, y eliminar la ambigüedad entre ellos utilizando una estrategia similar a la que se propone en esta tesis. Es en este contexto en el que esperamos que nuestro sistema sirva como punto de partida que, además de producir resultados de buena calidad, como se ha demostrado en este trabajo, tiene una estructura flexible que permite la implementación de nuevas ideas de forma mejorarlo.

La configuración final del sistema puede ser adaptada por el usuario de acuerdo con su necesidad, con el fin de lograr una mejor precisión, cobertura o F-measure.

Hemos implementado las metodologías propuestas en esta tesis en el proyecto Moara (cf. apartado F.1), una biblioteca de Java disponible para ser utilizado por la comunidad científica. Moara incluye clases que permiten al usuario probar el CBR-Tagger y el ML-Normalization que se describen aquí, e incluye la posibilidad de elegir los documentos de entrenamiento utilizados para entrenar el sistema y los métodos similitud de textos utilizados como características de máquina de aprendizaje. Dos versiones de la CBR-Tagger han sido integradas en U-Compare (cf. apartado F.2) y tenemos planes de integrar también el ML-normalización, ya que hay pocos sistemas para esta tarea integrados en esta plataforma.

También hemos propuesto una metodología para la extracción de las relaciones biomédicas con el razonamiento basado en casos. Hemos evaluado nuestra metodología en dos dominios, la extracción de relaciones entre enfermedad y el tratamiento con el corpus BioText (cf. apartado 3.5), y la extracción de eventos biomédicos con el corpus del BioNLP Shared Task (cf. apartado 3.4). Nuestros resultados muestran que el uso de RBC es factible para el problema de extracción de relaciones y que la metodología que aquí se propone obtiene resultados satisfactorios para los dos corpora.

El análisis de los errores presentados para la extracción de eventos biomédicos confirma la complejidad de las tareas, que incluye la extracción de los disparadores de eventos. Creemos que nuestro abordaje de aprendizaje automático es satisfactorio para esta tarea, pero más experimentos deben llevarse a cabo y otras características pueden ser consideradas para ambos clasificadores con el fin de mejorar el desempeño del sistema. Además, el análisis automático de los errores es una tarea difícil ya que ningún indicio es fornecido para los falsos positivos y falsos negativos por el sistema de evaluación.

En cuanto a las limitaciones de nuestra metodología para la extracción de relaciones, suponemos que las entidades han sido previamente extraídas en el texto para ambos dominios. Sin embargo, creemos que esto es una suposición razonable dado el rendimiento actual de los sistemas para reconocimientos de entidades. Además, y con el fin de reducir el tiempo de procesamiento, tenemos la intención de hacer cambios en la generación automática de candidatos de eventos mediante la inclusión de algunas limitaciones, y, en consecuencia, reducir el número de contextos de los candidatos que deben ser analizados por el clasificador de RBC.

En la extracción de relaciones, el contexto es esencial para la correcta solución del problema, especialmente cuando la especulación y la negación están bajo consideración. Una forma de explorar más el contexto de la frase es mediante el uso de un análisis profundo, que no ha sido explotado mucho más en nuestra metodología. Además, aproximadamente la mitad de los errores de falsos positivos y falsos negativos se debieron a los disparadores de eventos que no han podido ser

extraídos correctamente y en este caso, al igual que lo que se ha discutido para la extracción de genes y proteínas, otras características y una ventana más anchas de palabras podría ayudar a resolver este problema. Un poco de conocimientos del dominio, tal como un listado de los disparadores más frecuentes también podría ayudar y el mismo puede ser obtenido automáticamente a partir de los documentos de entrenamiento.

La extracción de eventos biomédicos es un buen ejemplo del nivel de dificultad en las tareas de extracción de relaciones. Sin embargo, ella contiene tipos de eventos que pueden ser fácilmente extraídos, tales como la expresión génica, cuya nomenclatura no varía mucho. Además, la expresión génica se compone sólo de un argumento, un tema, que es una proteína. Por el contrario, el evento "binding" puede estar compuesto de uno, dos o tres temas (proteínas), y es, por lo tanto, mucho más difícil de extraerse. Y aún más difícil son los eventos de regulación, que pueden tener proteínas u otros eventos como argumentos. Además, los eventos pueden tener modificadores tales como la especulación y la negación. En nuestra metodología, no hemos puesto mucho esfuerzo en la extracción de los modificadores. Además, la exploración del contexto de las frases y el uso de análisis sintáctico pueden mejorar el desempeño del sistema para este problema, así como para resolver co-referencias, que no hemos tenido en cuenta en esta tesis.

La metodología propuesta para la extracción de los eventos biomédicos todavía no está disponible en el proyecto Moara, pero también se ha integrada en la plataforma de U-Compare (cf. apartado F.2). Es parte de servidor común que permite una comparación de los resultados de algunos de los sistemas que participaron en la competición de BioNLP Shared Task (Kim, Ohta et al. 2009). También tenemos previsto evaluar nuestras metodologías con otros corpora. Una nueva versión de la competición de extracción de eventos ha tenido lugar durante 2010/2011 y nuevos documentos están disponibles, incluyendo algunos textos completos. También tenemos planes de probar nuestra metodologías para relaciones binarias, tales como la interacciones entre proteínas (Tikk, Thomas et al. 2010) y la interacción entre fármacos.

En cuanto a la aplicación Moara, también tenemos planes de hacer disponible una nueva versión con más funciones, especialmente el módulo de extracción de relaciones. Una mejor documentación de la biblioteca sería creada, así como una API Java. En cuanto a la tarea de extracción de relaciones, tenemos planes de implementar un modelo más flexible, con el fin de permitir que pueda ser utilizado para cualquier tipo de relación. Además, se podría poner a disposición algunas características que estarían listas para ser utilizadas en el abordaje de razonamiento basado en casos.

En general, el desempeño de la extracción de información biomédica es inferior en comparación con otros dominios, como por ejemplo el de noticias. Algunos autores (Zhou and He 2008) sostienen que una de la razones es que las ontologías y terminologías no son bien utilizadas o no son consideradas en absoluto, mientras que éstas son un requisito previo para la obtención de un buen desempeño del sistema. Para las metodologías que proponemos en esta tesis, este es un punto que debemos explorarse en el futuro, ya que pocas o ninguna ontología ha sido utilizada, con la excepción de la desambiguación en la normalización de genes y proteínas.

CHAPTER 2 INTRODUCTION

2.1 Motivation

Molecular biology is the discipline that study the biology at the molecular level (Lodish, Berk et al. 2000), i.e., important processes related to the living beings, such as the molecular structure, function and composition. It is related with both Biology and Chemistry sciences, and more particularly to biochemistry and genetics, and it is concerned to the understanding of the various systems of the cell, such as the Deoxyribonucleic Acid (DNA), Ribonucleic Acid (RNA), protein biosynthesis, metabolism and the way that these interactions are regulated in order to achieve a correct functioning of the cell.

Regarding other fields related to molecular biology, biochemistry focuses more on the role, function and structure of the bio-molecules. The latter may be defined as any organic molecule that may be produced by a living being, some large molecules, such as proteins and nucleic acids, and some small one, such as metabolites. On the other hand, genetics may be defined as the study of the difference among the various organisms due, for instance, to the absence of a gene. Genetics also includes the study of mutants, i.e., organism that lacks one or more components in relation to the normal phenotype, any observable characteristic of an organism, such as its morphology, development, behaviour, etc. Finally, Molecular Biology includes the study of the processes of DNA replication, transcription (or RNA synthesis) and translation of the RNA to amino acids.

Advances in biomedical technologies and high-throughput experiments such as DNA microarrays, gene expression (Holloway, van Laar et al. 2002) and mass spectrometry proteomics techniques (MacBeath 2002), next generation sequencing among others, are being widely used since the last decade and have enabled scientists to study biological systems from a global perspective. These new methodologies generate huge amounts of information related to genes and proteins at different levels. Therefore, the challenge lies in the ability to analyze and interpret this data, in which the bioinformatics community has made significant advances. However, these tasks are not trivial and the development of automatic methods is needed in order to assist in functional interpretation and extract interpretable facts and biological knowledge. This is one of the main challenges in bioinformatics research.

Besides, the comprehension of the complex biological complexes in eukaryotic organisms, such as human, inevitably needs the integration of all possible experimental data inferred from individual studies of some particular processes in distinct organisms. The integration of this information requires an accurate interpretation and analysis of many sources of information. Nowadays, there is no knowledge base in which this information may be completely found in a more structured way. The scientific literature is probably one of the richest sources of

information. For this reason, it is a great challenge for the scientific community to develop methods for extracting and then organizing this information automatically to allow its further analysis.

In the last decade, the interest for the biomedical text mining from clinical texts and the scientific literature has experienced a huge increment (Chapman and Cohen 2009). The main reason is that the literature covers all aspects of the biology, chemistry and medicine, there are almost no limits to the kinds of information that may be recovered through the use of an exhaustive and careful text mining. In this domain, biomedical text mining may be defined as the set of methods used to extract or retrieve knowledge which is hidden in texts and present it in a coherent way to be used by the biologists. Therefore, text mining is in charge of analyzing the texts in order to discover new information that would be hard to be retrieve in any other way.

In order to process and store this information, many are the computational methods that have been proposed in the areas of bioinformatics, computational biology and computer science, and specially, in the field of natural language processing (NLP), which may be defined as the group of methods for the automatic processing of documents written in natural language, such as in the English language. A system which performs text mining may be composed of one or more of the following tasks (Jackson and Moulinier 2002): information retrieval (IR), named entity recognition (NER), information extraction (IE) and knowledge discovery (KD). Each of these tasks is described below.

Information retrieval (Hersh 2008) is in charge of searching and retrieving documents that match some input information or query from a huge base of documents, such as the World Wide Web (WWW). In the biomedical domain, it can be described as a way to gather relevant texts, usually scientific documents from the PubMed[1] database (Fenton and Williams 2005) (cf. A.1). It is usually the first step in any text mining system. Information retrieval can also be performed inside a text in order to decide which part of the document is more relevant according to the user's query.

Named entity recognition (Park and Kim 2006) is the identification of some predefined entities in a determined text, which may have been acquired in the information retrieval step or in any other way, such as manually. For the biomedical domain, the entities are usually genes, proteins, diseases, drugs, anatomical parts, etc. Sometimes, more than one type of entities is extracted from the text at the same time and a categorization is needed in order to define the type of entity for each of the extracted mentions. Related to this step is the named entity normalization, which is the association of each mention to one name or identifier

[1] http://www.ncbi.nlm.nih.gov/pubmed

in a predefined ontology or terminology. This step may be performed together with named entity recognition or separately.

Information extraction tries to extract predefined relationships among some entities from the text of an unstructured document. The entities must be given or previously extracted in the named entity recognition step. For the biomedical domain, the protein-protein interaction (Krallinger, Leitner et al. 2008) is one of the more popular information extraction tasks. This information extraction task is growing in importance particularly due to the increasing interest in system biology (Ananiadou, Pyysalo et al.).

Finally, knowledge discovery tries to discover hidden or implicit information from the texts. This is usually performed by proposing some potential hypothesis derived from the information extracted in the previous step. For example, by trying to infer indirect relationships, a knowledge discover system may generate a hypothesis that A is related to C if the text describes that A is related to B and B related to C.

The rising interest on biomedical text mining is related to the growth and accumulation of scientific literature and the fast process of discovering biomedical information. The computational methods used for the processing of biomedical literature allow the easy and fast access of biologists, bioinformatics and database annotators (curators) to the relevant texts. However, most of the time, the information extraction task is carried out manually. The information is extracted from the relevant scientific publications and further stored in huge databases available, which are usually freely to the community, such as Entrez Gene (Maglott, Ostell et al. 2007) and Uniprot (2009) for genomic information, and MINT (Chatr-aryamontri, Ceol et al. 2007) and IntAct (Kerrien, Alam-Faruque et al. 2007) in the domain of protein-protein interaction. These freely available databases are of great importance as the results of the distinct experimental and bioinformatics methods are usually interpreted by using the information they contain. Nowadays, it is unconceivable to try to understand the complex biological processes in some determined conditions in complex organisms, such as human, without taking into account all the information that might be summarized for similar processes from the scientific community.

Unfortunately, all this information is not always found in a structured format such as relational databases that are of easy access to the researchers. On the contrary, most of this information is found in a unstructured format in bases of free text or poorly structured documents, such as PubMed (Fenton and Williams 2005). Due to this situation, a great number of research groups in the both areas of molecular biology and computer sciences have dedicated great efforts in the development of new methodologies to extract huge amounts of information from the scientific literature. This is the scope of the work described here.

2.2 Contribution

This thesis proposes new methodologies to solve several state-of-the-art biomedical text mining problems. When described individually, these tasks seem to be unconnected, but altogether, they are necessary steps in the automatic process to extract knowledge from biomedical literature. In particular, this thesis focus on named entity recognition, relationship extraction and entity mention normalization. Our detailed contribution and an outline of the thesis are described below.

Chapter 3 begins with an introduction to natural language processing, such as tokenization, part-of-speech tagging and syntactic parsing. They are followed by the metrics used in the evaluation of the methodologies proposed here. Finally, we describe the state of art for the following tasks: recognition and normalization of gene and protein mentions and extraction of biomedical relationships.

Chapter 4 describes in details the methodologies we propose for the tasks of recognition of named entities, and particularly for the extraction of genes and proteins and event triggers. We also describe our methods for the information extraction, i.e., relationships such as biomedical events and disease-treatment. The methodologies we propose use the case-based reasoning approach. Evaluation and results are presented for each of the tasks.

Chapter 5 describes our methods for the normalization of the genes and proteins mentions to their identifiers. Our approaches are based on dictionary lookup and machine learning algorithms and they include the disambiguation of the identifiers, when necessary. Evaluation and results are also presented for each approach.

Chapter 6 presents the conclusions of this thesis and discuss the future work which are both in progress or that we plan to carry out for the continuity of the methodologies and the softwares which have been developed during the thesis.

The development of new methods for information retrieval and knowledge discovery was beyond the scope of this work and was not included. Instead of performing retrieval of relevant documents, we have used state-of-art corpora for the evaluations of each of the approaches proposed here.

In the appendix we described the materials that have been used for the development of the algorithms and their evaluation. It includes databases and terminologies, details on the corpora used in the evaluation and a short description of the external libraries which have been used in the developed of our systems and methodologies. The appendix also includes the functionalities of our system which are available for use as part of the Moara project: CBR-Tagger, ML-Normalization and BioEvent Extractor. Finally, we list the publications which are related to this thesis, both in journals or conferences and workshops.

CHAPTER 3 TEXT MINING

In this chapter we review some concepts and methods we use as building blocks for the methodologies we propose and which are referred throughout in this thesis. We start describing some concepts related to the natural language processing (cf. 3.1), such as sentence splitting, tokenization and parsing. They are common pre-processing steps present on most of the text mining systems. Then we proceed with the description of the tasks for which we propose our methodologies, namely: named-entity recognition (cf. 3.2), relationship extraction (cf. 3.3) and entity mention normalization (cf. 3.4). Finally, we present the evaluation metrics we use for the assessment of our methods for these three tasks (cf. 3.5).

3.1 Natural Language Processing

When performing natural language processing (Jackson and Moulinier 2002; Hahn and Wermter 2006), some pre-processing steps are usually necessary to be carried out in the text. In this section, we describe some of these pre-processing steps.

3.1.1 Sentence splitter

One of the first operations that are usually carried out in textual documents is the delimitation of the sentences that constitute the text, using a sentence splitter. This step is usually necessary as most of the text mining methodologies are still limited to a single sentence. This is due to the difficulties in dealing with anaphora, i.e., references to the same entity (e.g. a gene) using different names (e.g., the gene name and the pronoun “it”). Although it seems an easy task when considering the period as separator, a period may be ambiguous and may denote, for instance, a decimal point or an abbreviation. In this thesis, Lingpipe sentence splitter has been used (cf. C.1), for example, for separating the sentences in the datasets which were used for the extraction of the biomedical events (cf. 4.4).

3.1.2 Tokenizer

The next step in a natural language processing pipeline is usually the use of a tokenizer, which separates the sentence into tokens. This is usually necessary as most of the text mining methods consider the token as the smallest unit in the text. They are also building blocks for some text structures, such as the window of tokens (cf. 4.2.1.1). Here again, it could seem a straightforward task, by separating the tokens according to white spaces. However, it might not be so obvious depending on the language under consideration. Sometimes hyphens may also serve to delimitator for the tokens. In contrast, commas and periods may not, as for instance when they are part of a decimal number. Tokenizers have been widely used in this work for separating the tokens, sometimes implemented manually as for the gene and protein recognition task (cf. 4.6) or by using an external library, such as Lingpipe (cf. C.1), for the extraction of biological events (cf. 4.4) and for the extraction of relationships between diseases and treatments (cf. 4.5).

3.1.3 N-grams

N-grams may be defined as a subsequence of phonemes, syllables, tokens or letters, depending on the case. They may be used as the smallest unit in the text, instead of the tokens (cf. 3.1.2). In case of subsequences of size 1, 2 or 3, they may be denoted as unigram, bigram or trigram, respectively. In this work, we usually use bigrams and trigrams when referring to subsequences of letters in a token. For instance, the word "n-myristoyl transferase" might be represented by the trigrams {n, myr, yri, ris, ist, sto, toy, oyl, tra, ran, ans, nsf, sfe, fer, era, ras, ase}. In this work, bigrams and trigram have been used as one of the features of the machine learning algorithms developed for the gene and protein normalization task (cf. 5.4).

3.1.4 Part-of-speech tagger

One of the most used syntactic analyzer tools is the part-of-speech (POS) tagger which associates each token in the text to its syntactic tag. Therefore, it indicates whether the token is a noun, a verb or an adjective, for instance. The importance of a POS tagger lies in the syntactic disambiguation of each word in the text. For instance, the word "fish" might refer to a noun or a verb depending on the context. The POS taggers may be rule-based, i.e., given by a set of lexical and contextual rules, or statistical-based, i.e., by assigning to a token the tag with the highest probability given the previous "n" tokens. Manually annotated corpora may help in the training of a POS tagger, as for instance the general purpose Penn TreeBank corpus (Marcus, Santorini et al. 1993) and the GENIA corpus (Ohta, Tateishi et al. 2002) for the biomedical domain. The POS tags have been used as one of the features for the case-based reasoning algorithms developed for the extraction of biological events (cf. 4.4) and for the extraction of relationships between diseases and treatments (cf. 4.5), which were extracted using the Stanford parser (cf. C.4). Figure 3.1 shows an example of the POS tags outputted by the Stanford parser for the example sentence provided in its demo page[2].

3.1.5 Stemmer and Lemmatizer

A stemmer associates morphological variants of a term to the root form. A morphological variation of a word may be of the inflectional or derivational type. The first one is related to variations of a word without changing its part-of-speech tag, usually expressing singular/plural and past/present tense variations. For instance, "fish" and "fishes" are inflectional variations of the root "fish". On the other hand, the derivation morphology refers to variations with different part-of-speech tags, i.e., different syntactic function in the sentence. For instance, "fisher" (noun) and "fishing" (verb) are derivational variations for the root "fish". The lemmatizer is a bit more complex tool which groups together the different variations of a word, usually by performing dictionary look-up and by understanding the context, as for instance, making use of the part-of-speech tags.

[2] http://nlp.stanford.edu:8080/parser/

This is the main difference between a lemmatizer and a stemmer, as the latter is based exclusively on simple rules, for instance, by removing the "ing" suffix for verbs in the continuous tense. One example of the advantage of the lemmatizer over the stemmer is for example the word "better" that would have "good" as its lemma. This association would be missed by a stemmer as it does not perform a dictionary look-up.

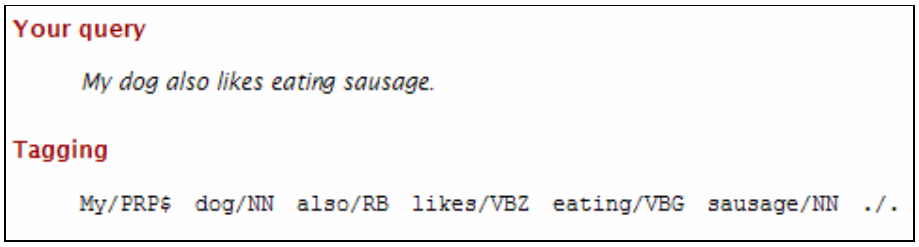


Figure 3.1: Example of the part-of-speech tags for a sentence.
Output of the Stanford parser for the sentence "My dog also likes eating sausage". Regarding the POS tags, "PRP$" stands for a pronoun, "NN" for a noun, "RB" for and adverb, "VBZ" for third person singular present verb and "VBG" for a verb in the continuous tense.

The importance of using lemmas and stems as features for a text mining method is that it reduces the number of features to be dealt, as different words are simplified to their common roots. It also allows the system to learn the different variation of the same root, such as the different tenses of a verb. One of the widely used stemmers is the Porter (cf. C.2) that has been used in this thesis in the as part of the disambiguation step of the gene and protein normalization task (cf. 5.5). Lemmatizers have also been used in this work, such as the one available in the Dragon toolkit (cf. C.3) for the extraction of biological events (cf. 4.4) and for the extraction of relationships between diseases and treatments (cf. 4.5).

3.1.6 Chunkers and Parsers

The chunkers identify special phrasal units, such as noun, verbal or prepositional phrases from a text, usually making use of both lexical and part-of-speech tags (cf. 3.1.4). These tools usually rely on annotated corpora for the training step and they have been proven beneficial for some tasks, such as named-entity recognition, as most named entities are contained in noun and preposition phrases. Figure 3.2 shows an example of the chunk tags outputted by the GENIA tagger for a given sentence[3].

As shown in Figure 3.2, the chunks usually come with an extra tag which indicates the position of the respective token in the phrase. The most common formats are the BIO and BIEWO. The BIO format is composed of three tags which indicate the first token of the phrase (B), the following ones (I) and tokens which are not part of

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

any phrase (O), usually punctuation marks. These tags may also be used for the named-entity recognition (cf. 3.2), in order to indicate the position of a token in the mention, when it is composed of more than one token. In this work, the chunks have been used as one of the features in the case-based reasoning method developed for the recognition of biological events (cf. 4.3.1).

```
Inhibition      Inhibition      NN      B-NP
of              of              IN      B-PP
NF-kappaB       NF-kappaB       NN      B-NP
activation      activation      NN      I-NP
reversed        reverse         VBD     B-VP
the             the             DT      B-NP
anti-apoptotic  anti-apoptotic  JJ      I-NP
effect          effect          NN      I-NP
of              of              IN      B-PP
isochamaejasmin isochamaejasmin NN      B-NP
.               .               .       O
```

Figure 3.2: Example of the chunks for a sentence.
Output of the GENIA parser for the sentence "Inhibition of NF-kappaB activation reversed the anti-apoptotic effect of isochamaejasmin.". The first column lists the tokens, the second the base form of the tokens, the third the part-of speech tags and the composition of a BIO tag and the chunk tag. Regarding the chunk tags, "NP" refers to noun phrase, "VP" to verbal phrase and "PP" to preposition phrase.

A parser performs a syntactic analysis of a text through the identification of the clauses, i.e., word sequences that represent a subject or a predicate. The parsing of a text is done with respect to a grammar, i.e., a set of rules that specify which combinations of part-of-speech tags generate well-formed phrases and sentence structures. The parsers may be classified into two approaches (Miyao, Sagae et al. 2009): dependency parsing and deep parsing. The chunkers and the dependency parser may also be called shallow parsers as their objective is to extract syntactic information efficiently by sacrificing the integrity of the analysis. On the other hand, a deep parser analyzes the whole sentence, it is much more complex but also more precise. The two approaches will be described in details below.

A dependency parser compute a tree structure of a sentence where the nodes represent the words and the edges represent relationships between words. The main advantage of the dependency trees is that they are a reasonable approximation of the semantics of the sentences and are easy to be used by an NLP application. Figure 3.3 presents an example of the dependency tags outputted by the Stanford parser[4] (cf. C.4) and two representations of the tree structure.

From the tree structure showed in Figure 3.3, it is possible to infer some important features that may be especially helpful for the relationship extraction task (cf. 3.3). For example, it can be noticed that the word "association" is the root of the tree

[4] http://nlp.stanford.edu:8080/parser/

and this word is a strong indication of the association between the infection of the human papillomavirus and the nonsmoking lung cancer. Other important definition here is the parent of a node, for example the node "papillomavirus" in respect to the "Human" node and the children of a node, for example, the nodes "papillomavirus", "(HPV)", "infection", "is" and "with" are the children of the node "association". Also important is the information given by the least common subsumer (LCS), the lowest level node that is parent of two given nodes. For example, the node "association", besides being the root of the tree, it is the least common subsumer of the nodes "infection" and "cancer". In this thesis, the dependency parser and some of the above feature of the tree structures has been used in the recognition of the trigger (cf. 4.3.2) as part of the extraction of biological events.

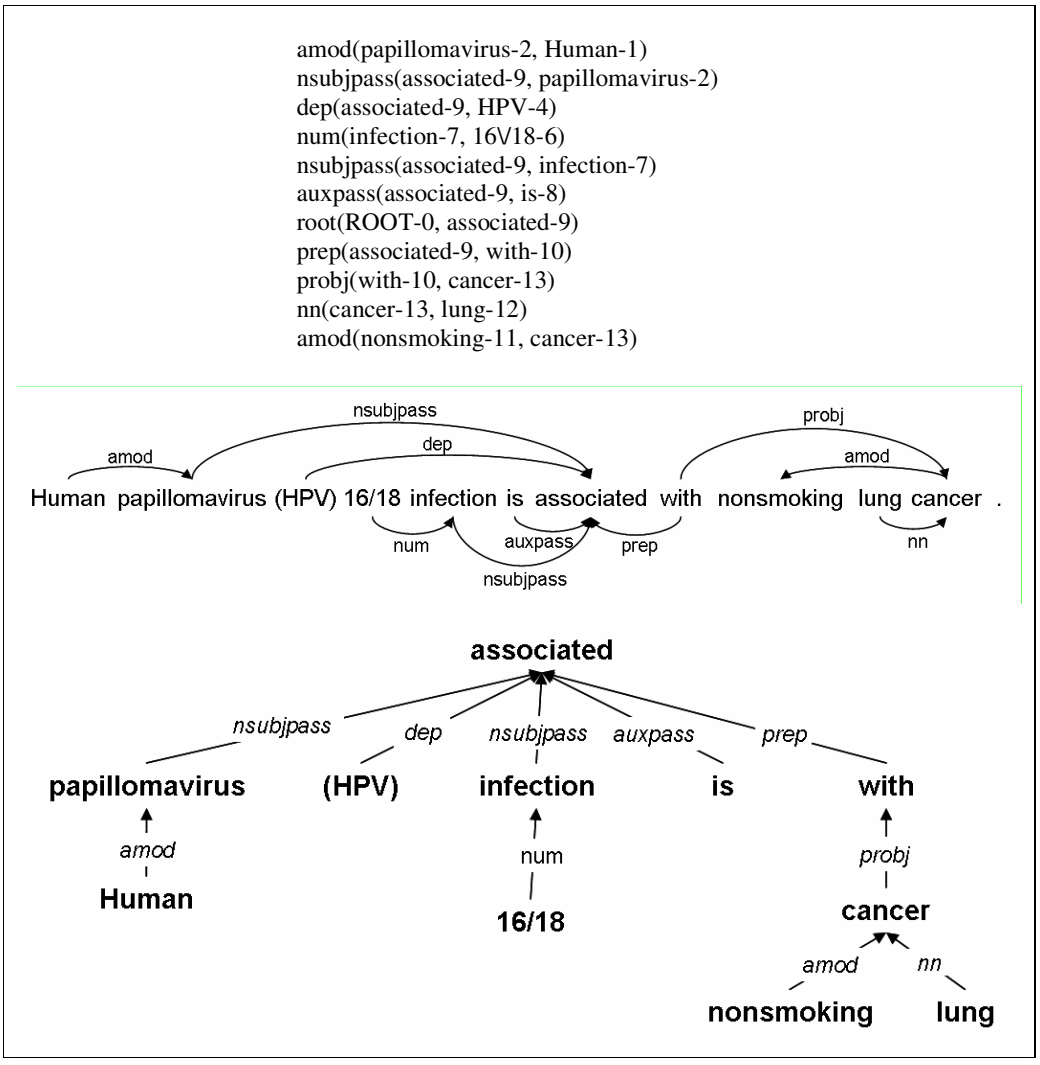


Figure 3.3: Example of the dependency tags for a sentence.
Output of the Stanford parser for the sentence "Human papillomavirus (HPV) 16/18 infection is associated with nonsmoking lung cancer." The syntactic structure of the sentence is also presented in terms of its dependency tags and as a tree.

A deep parser outputs a phrase structure tree based on the Penn TreeBank style. Although they are usually based on probabilistic context-free grammars (PCFGs), the parameterization of the probabilistic model may vary depending on the parser. Figure 3.4 presents a parser tree outputted by the Stanford parser (cf. C.4).

```
(ROOT
  (S
    (NP (NNP Human) (NNS papillomavirus))
    (PRN (-LRB- -LRB-)
      (NP (NNP HPV))
      (-RRB- -RRB-))
    (NP (CD 16\/18) (NN infection))
    (VP (VBZ is)
      (VP (VBN associated)
        (PP (IN with)
          (S
            (VP (VBG nonsmoking)
              (NP (NN lung) (NN cancer)))))))
    (. .)))
```

Figure 3.4: Example of the Penn TreeBank output of the Stanford parser.
Output of the Stanford parser for the sentence "Human papillomavirus (HPV) 16/18 infection is associated with nonsmoking lung cancer." (PubMed identifier 20578176). The syntactic structure of the sentence is presented as a tree and the Penn TreeBank tags.

A deep parsing computes in-depth syntactic and semantic structures based on syntactic theories, such as Head-Driven Phrase Structure Grammar (HPSG). Deep parsers also generate predicate argument structures (PAS) which are able to express deeper relationships, such as long distance dependencies. In thesis, we have not made use of deep parsing in any of our methods.

3.2 Named Entity Recognition

Named entity recognition in text documents is the identification of some predefined types of entities, such as people, organization or places, in a given text (Park and Kim 2006). As a result, a named entity (or a mention) may be defined as phrase composed of one or more tokens that denote a specific object, such a person, organization or city. In the biomedical domain, these entities are usually genes, proteins, diseases, cell lines, etc. The extraction of these entities is a preceding step for other text mining tasks, such as the retrieval of documents related to a certain gene or protein, or the initial characterization of the proteins involved in a text for a system in charge of the extraction of protein-protein interactions or gene expression events, for example.

Ideally, a named entity recognition system should also be able to detect variations of the original entity, such as abbreviations, plurals and compounds. Also, it should include the identification of anaphoric expressions, such as "it" or "the protein",

which refers to entities that have been previously introduced in the text. However, this is still a hard task in text mining and it might require the use of high-level natural language processing in order to obtain a syntactic structure of the text under consideration.

The named entity recognition task includes the exact identification of the mention in the text, i.e., its boundaries, which are defined by the position of the first and last characters that delimitate its component tokens. This recognition is not an exact task as different experts can annotate the same text in different ways. Disagreement among the annotators may be related to the existence or not of a certain entity or regarding its boundaries. Therefore, entities consisting of more than one token can be annotated in different ways and may have alternative synonyms. For instance, a certain annotator may consider that only part of the tokens is enough, while another would prefer to include some more tokens to its complete characterization. As an example of NER, Figure 3.5 shows an example of a document where different classes of chemicals have been identified.

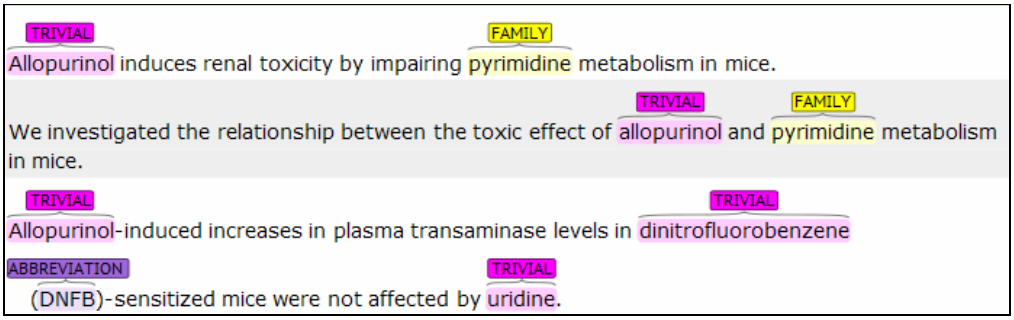


Figure 3.5: Examples of named-entity annotations for chemicals.
The chemical annotations are from the SCAI corpus[5] for the document 10823345 which is available in Stav visualization tool[6].

An entity can be identified in many ways in a text. The easiest one is to identify it using a special tag. For example, in the training documents of the BioCreative 1 task A (Smith, Tanabe et al. 2008), genes and proteins were identified with the "NEWGENE" tag. Sometimes, an additional label "NEWGENE1" was used for consecutive mentions, in order to identify the boundaries of each of them. However, this tag does not distinguish whether the mention is composed of more than one token and it does not identify in a special way the first and the last token of the mention, for example.

A format often used by some authors (Tsuruoka and Tsujii 2005; Chen, Liu et al. 2007; Dai, Hung et al. 2007; Ganchev, Crammer et al. 2007; Huang, Lin et al.

[5] http://www.scai.fraunhofer.de/en/business-research-areas/bioinformatics/research-development/information-extraction-semantic-text-analysis/named-entity-recognition/chem-corpora.html
[6] http://corpora.informatik.hu-berlin.de/

2007) is the IOB2 (Tjong Kim Sang and Veenstra 1999), which provides basic information in regard to the number of words in the mention and its boundaries. This format is composed by the "B", "I" and "O" tags, which represent the first token of the mention, the second and the forthcoming one, and the tokens that do not belong to mentions, respectively. Another format which is quite similar to the IOB2 is the IOE2, which does not include the "B" tag and uses the tag "E" to designate the last token of the mention. Since there is no start tag, the "E" tag also identifies a reference composed of a single word. Comparative studies of the influence of the format representation in the final results of a NER approach can be found in the work of (Tjong Kim Sang and Veenstra 1999).

Finally, we introduce the more complete BIEWO format which according to (Vlachos 2007) produces better results than the use of the IOB2, for example. This format is composed by five tags, as follows: "B" for the first token of the mention, in case of more than one; "I" for the intermediate token, in case of more than two; "E" the last token, in case of more than one; "W" for mentions composed of a single token; and "O" for any token outside the mention. Regardless of the format used in a certain corpus, it is usually possible to make the conversion from one format to another, depending on the specific needs of the methodology and the algorithm in use. However, a study of various representations of formats (Tjong Kim Sang and Veenstra 1999) has shown that the format has little influence on the final results. Section 3.2.1 below will describe in details the previous work related to the recognition of gene/proteins mention, which is one of the focus of this thesis.

3.2.1 Recognition of gene and protein mentions

The main difficulties regarding the recognition of genes and proteins are the existence of a large number of such entities, the lack of rules concerning their nomenclature and the resistance of the scientific community to use the existing ones (Tamames and Valencia 2006). Despite the existence of curated genomic databases, such as HUGO (Povey, Lovering et al. 2001) for the human and Uniprot (2009) for proteins, the gene/protein nomenclature suffers of various issues, such as ambiguity, synonyms and variations.

Regarding the ambiguity, different entities may share the same name and some nomenclature may even coincide with common English words (e.g., "deafness"), which complicates even more their detection in a text (Leser and Hakenberg 2005). In addition, newly-discovered entities sometimes are assigned to a name that is already in use for an existing gene or protein. The ambiguity of an entity may even occur in relation to a different type of entity, for example, a gene name may coincide with a cell line name.

Additionally, a gene or protein may also have more than one name, usually called synonyms or aliases. This issue makes the construction of a complete list of

gene/protein synonyms for a certain organism a much harder task, even with the help of experts. Sometimes, not even specific databases for a certain organism are able to maintain such a complex and dynamic list of synonyms. Finally, variations are also very common in the nomenclature of genes and proteins, and all these variations should ideally be mapped to its standard name. These are the most common variations in gene/protein names:

- character-level variations: presence or absence of special characters, such as hyphens, parenthesis, upper cases, commas, etc;
- word-level variations: due to the use of synonyms as part of the name, such as "gastric" and "stomach";
- word-order variations: when the same tokens appear in a different order;
- acronyms, e.g., the use of "TNF" instead of "tumor necrosis factor".

The gene/protein recognition task is often used as a prelude to other problems, such as the normalization of genes and proteins. The latter becomes an easier task if the mentions available in the text are provided, as well as their exact localization in the text. Additionally, the use of context information, i.e. the tokens or sentences nearby the mentions, may facilitate the normalization task (cf. 3.4).

The approaches for the recognition of gene and protein mentions may be classified in methods based on a dictionary of synonyms, manual rules and machine learning. Due to the time it takes to register a new gene or protein synonym, approaches that are based on pure dictionary look-up fails to recognize names that are not yet included in the genome databases. Also, the variations on the nomenclature are not always included in these lists. Therefore, some of the most successful methodologies, for example (Hanisch, Fundel et al. 2005), are based on a initial list of synonyms that are further manually and automatically expanded with some variations.

Many solutions have been proposed for the gene/protein recognition task (Smith, Tanabe et al. 2008). The methods range from rule-based systems (Fukuda, Tsunoda et al. 1998) to string approximate matching (Krauthammer, Rzhetsky et al. 2000). The vast majority of systems which have obtained good results in this task have used conditional random fields (CRF) (Lafferty, McCallum et al. 2001) in their implementation, a conditional probability method widely used for the label and classification of data. Some example of systems based on this methodology are the works of (Dai, Hung et al. 2007) and (Katrenko and Adriaans 2007), as well as ABNER (Settles 2005) and BANNER (Leaman and Gonzalez 2008), two of the most widely used gene/protein recognition tools.

Other methodologies include bidirectional inference used for the development of the GENIA tagger (Tsuruoka and Tsujii 2005), support vector machines (SVM), as

in the work of (Chen, Liu et al. 2007) and (Huang, Lin et al. 2007), semi-supervision learning (Ando 2007), which obtained the best results in the BioCreative 2 Gene Mention task (Smith, Tanabe et al. 2008), case-based reasoning (CBR) (Neves, Carazo et al. 2010) and an ensemble of many machine learning algorithms (Naïve Bayes, k-nearest neighbor, AdaBoost with Naïve Bayes and C4.5 decision trees) (García, Puertas et al. 2007). Also, the work of (Zhou, Shen et al. 2005) makes use of several different methods and unified them using a feedback system.

Some systems include a post-processing step in order to improve the results obtained in the previous stage. One example of this kind of procedure is the verification of the boundaries of the mentions (Chen, Liu et al. 2007), in order to check inconsistencies or redundancies, such as whether any important word (part of the mention) has not been recognized, or whether the tags are correct assigned, in case of using the BIO or the BIEWO format. The post-processing procedure may also consist of simply checking the open-close pairs of parentheses or brackets (Kuo, Chang et al. 2007). Finally, some systems may also include specific procedures for checking abbreviations and their corresponding full names. Such approaches consist of adding or removing tokens from the mention until the abbreviation matches the full form (Chen, Liu et al. 2007).

When using machine learning algorithms for the automatic recognition of gene/protein mentions, some features must be provided for the characterization of the text. Some examples of such features are listed below:

- the length of the mention (Huang, Lin et al. 2007);
- whether the mention is a long term (full name) or an abbreviation (short name);
- the position of the word in the sentence, e.g., whether it is at the end of it (García, Puertas et al. 2007);
- orthographic features (Klinger, Friedrich et al. 2007), such as upper and lower cases (in distinct positions in the mention), presence of special characters (parentheses, hyphens, etc.), numbers (integers, floats or Roman), Greek letters (Neves 2007), among others;
- an indicative whether the reference occurs within quotation marks or brackets (Dai, Hung et al. 2007);
- the use of a stem or lemma of the mention and the surrounding tokens (Vlachos 2007);
- Part-of-Speech tags (Klinger, Friedrich et al. 2007) (Finkel, Dingare et al. 2005);
- suffixes and prefixes composed of two to four characters (Chen, Liu et al. 2007) for the identification of words from the biological environment;

- n-grams, usually bigrams or trigrams, of the mention (Struble, Povinelli et al. 2007);
- presence of specific biomedical terms (Struble, Povinelli et al. 2007) (Ganchev, Crammer et al. 2007), by exact or approximate match, usually using a lexicon of terms;
- presence of the three-letter symbols that represent amino acids or nucleotides (Kuo, Chang et al. 2007);
- classification of the part of the mentions according to some predefined classes (Tamames 2005), for instance, "keyword", "stopword" and "location".
- any of the features described above for a delimited window of context (Huang, Lin et al. 2007), i.e., some predefined number of words that comes before and/or after the mention;

Regarding the order of the words in the text, it can be represented by reading from left to right (forwards) and reverse, from right to left (backwards). In some of our experiments, we have considered both directions (Neves, Chagoyen et al. 2008). According to (Kuo, Chang et al. 2007), the backwards direction performs better than the forward one for the gene/protein recognition, both regarding precision and recall.

Some authors have also used algorithms for the selection of the best features to be used by the classifier, such as the Sequential Forward Selection (SFS) in the work of (Dai, Hung et al. 2007). The selection of the most important features is a important issue regarding the disk space and the time required for training and testing the system. Additionally, some methods have been proposed for reducing the number of features under consideration and for improving the performance of the system without sacrificing the quality, such as the numerical normalization in (Dai, Hung et al. 2007), in which entities of the same family are reduced to a common term. For example, the terms "interleukin-2" and "interleukin-3" are reduced to "interleukin-1."

3.3 Biomedical Relationship Extraction

In natural language processing, information extraction (IE) can be defined as the task of finding essential information from a textual document. This data can be related to one or more predefined entities and the representation of the relationship among these entities is usually carried out by filling up the slot of a template (Ananiadou and Nenadic 2006). More specifically, the biomedical relationship extraction is a specialization of the information extraction task in which the entities are related to the medical or the molecular biology domains, such as genes, proteins or diseases.

Relationship extraction is a key issue in text mining as it takes part in many biological processes, and many efforts have been dedicated to this matter. As an

example, online databases are available for the storage of interaction between pairs of proteins, such as the Molecular INTeraction Database (Chatr-aryamontri, Ceol et al. 2007) and IntAct (Kerrien, Alam-Faruque et al. 2007). Most of the data contained in these databases have been curated manually by experts.

By far, the most popular task in the biomedical text mining is the protein-protein interaction (PPI) (Krallinger, Leitner et al. 2008) (cf. Figure 3.6), and more recently the extraction of biomedical events (Kim, Ohta et al. 2009), both of them due to challenges which have taken place in the last years. These challenge initiatives, and consequently, the availability of annotated corpora, have increased the number of solutions for the extraction of biomedical relationships as well as the improvement the results. For instance, the BioCreative II protein-protein interaction task (Krallinger, Leitner et al. 2008) consisted of four tasks, including the extraction of PPIs in full-text documents.

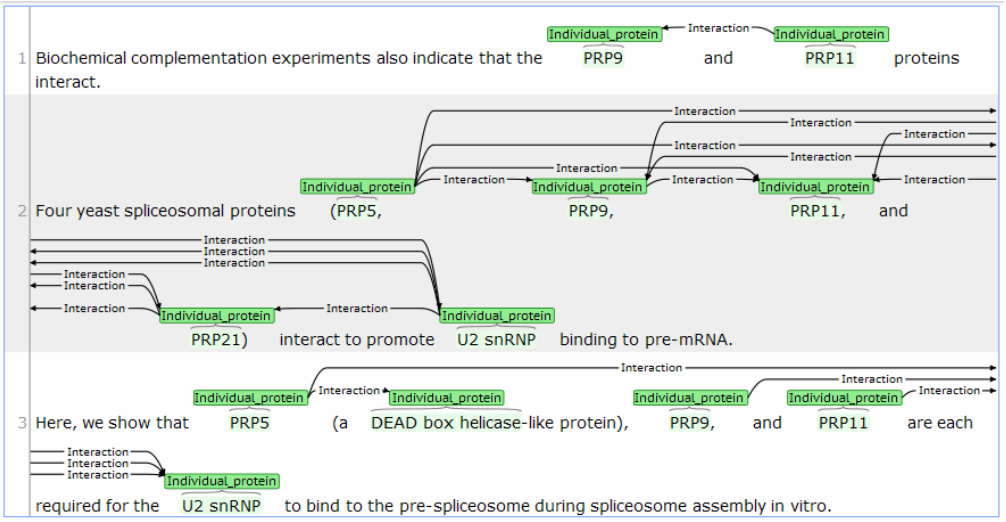


Figure 3.6: Example of protein-protein interactions.
The example comes from the document d100 of the BioInfer corpus[7], which is available in Stav visualization tool[8].

More recently, the BioNLP Shared Task of Event Extraction (Kim, Ohta et al. 2009) has proposed the identification of a variety of biomedical events. Some of the events were easier to extract, such as the gene expression or the phosphorylation, while some other were more complex, such as binding and regulation. Additionally, the identification of negations and speculations made the tasks even more complex. The f-measure of the best participating solutions has ranged from about 40% (negative regulation event) to almost 80% (gene expression event). Some examples are shown in Figure 3.7.

[7] http://mars.cs.utu.fi/BioInfer/
[8] http://corpora.informatik.hu-berlin.de/

Regarding corpora annotated with biomedical relationships, we cite initiatives such as GENIA (Kim, Ohta et al. 2008) , GREC (Thompson, Iqbal et al. 2009), the BioCreative PPI corpus (Krallinger, Leitner et al. 2008) and theunified format for five PPI corpora (Pyysalo, Airola et al. 2008). These corpora have allowed the development of many solutions based on supervised learning algorithms.

Relationship extraction is a higher level task which usually depends on the good performance of some low level tasks, such as tokenization (cf. 3.1.2), sentence splitter (cf. 3.1.1) or part-of-speech tagging (cf. 3.1.4), and other high level tasks, such as named-entity recognition (cf. 3.2). The extraction of relationships has lead the biomedical text mining community to the use of more full texts instead of only abstracts as the latter is usually very poor in relationships.

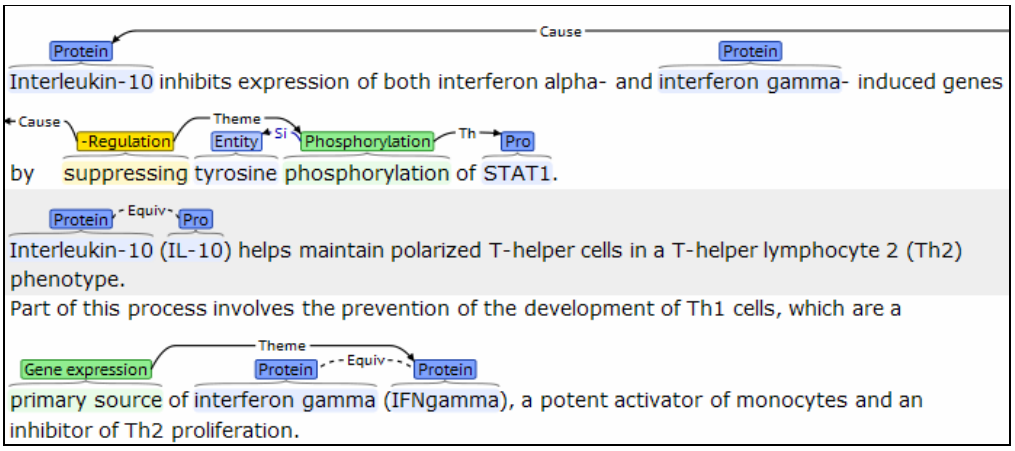


Figure 3.7: Complex biomedical relationships from the BioNLP task.
The biological events come from the training dataset (document 10029571) of the BioNLP Shared task[9], which is available in Stav visualization tool.

Regarding the methods used for the extraction of biomedical relationships, (Zhou and He 2008) classify them into three classes: co-occurrence, pattern matching and machine learning. However, other authors (Faro, Giordano et al. 2011) consider only two approaches, co-occurrences and natural language processing. The first approaches which have been proposed for relationship extraction relied in co-occurrence and pattern matching. Later, more complex natural language processing methods have been used in order to deal with complex relationships. Additionally, the use of NLP allows a better understanding of the context in order to take into account, for instance, negation and speculation. Hybrid approaches which combine more than one of these approaches have also been proposed (Tikk, Thomas et al. 2010). Each of the approaches proposed for the extraction of biomedical relationships is described below.

[9] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/

Co-occurrence considers that entities which appear together in the same text, usually in the same sentence, are supposed to be related. It has been used for extraction binary relationships, such as gene-disease (Tsuruoka, Tsujii et al. 2008). However, this approach is unable to identify negation or speculation, as only the presence of the entities are considered, but not the context in which they appear. Additionally, this approach is not very precise, as sometimes, unrelated entities may appear in the same sentence.

The rule-based approach uses some manual rules or patterns to define the possible relationships, usually within a sentence. This method may be used in conjunction to some statistical methods in order to estimate the confidence of a relationship. Although the use of manual rules may provide good results, such as the work of (Kilicoglu and Bergler 2009), heavy manual effort is need in order to build the patters and the resulting system cannot be easily adapted to other domains.

The computational linguistics approach analyzes the syntax and semantics of a text in order to obtain relationships among the predefined entities. Pre-processing of the text usually includes tokenization (cf. 3.1.2), part-of-speech tagging (cf. 3.1.4) and syntactic parsing (cf. 3.1.6). However, parsing unrestricted biomedical text may be extremely difficult and the performance of the parsing has a direct influence on the performance of this approach. Usually, this approach is used only within a sentence. In order to be abe to detect relationships across sentences, a system needs to deal with anaphoras.

Shallow parsers (cf. 3.1.6) allow the identification of coordinating conjunctions and negation and they usually perform well for extracting simple binary relationships (Giuliano, Lavelli et al. 2006). However, their performance decreases considerably for more complex relationships and relational clauses. By using deep parser (cf. 3.1.6), more precision can be achieved and complex relationships may be identified, usually unable to be recognized by the shallow parsers. In the BioNLP Event Extraction shared task, the system which performed better were the ones which made some use of deep parsers (Kim, Ohta et al. 2009). Other studies in this field have also suggested that the use of parsers (Miyao, Sagae et al. 2009) does improve the performance of the extraction of relationships between entities. The problem of using deep parsers is their high complexity and computational effort. Other computational linguistic methods have also been used in the work of (Yakushiji, Tateisi et al. 2001) and in the RelEx system (Fundel, Kuffner et al. 2007) for the extraction of a variety of relationships. This approach is usually used combined with the machine learning algorithms.

Finally, one or more machine learning algorithms may be used to infer the relationships without the need of defining a set of rules or grammar. However, a collection of documents precisely annotated with the relationships and its

corresponding entities is usually necessary. A variety of machine learning algorithms have been used in the BioNLP Event Extraction shared task, such as C4.5 (Móra, Farkas et al. 2009), support vector machines (Bjorne, Heimonen et al. 2009) and case-based reasoning (CBR) (Neves, Carazo et al. 2009), as proposed in this thesis (cf. 4.4). A similar solution to CBR is the memory-based algorithm implemented by (Morante, Van Asch et al. 2009).

3.4 Entity Mention Normalization

3.4.1 Normalization of gene and protein mentions

The normalization of biological entities, also known as automatic term recognition (Ananiadou and Nenadic 2006), includes not only the identification of named-entities in the text, but also the association of each mention to its unique identifier in a specific database or ontology. Figure 3.8 and Figure 3.9 illustrate a text in which some gene and protein mentions are normalized to their identifiers.

Eukaryotic cells localize selected mRNAs to a region of the cell as a means to sequester proteins. Signals within the 3' untranslated region (3' UTR) facilitate mRNA localization by both actin and microtubule cytoskeletal systems. Recently, an mRNA in the yeast Saccharomyces cerevisiae, **ASH1 <S000001668>**, was shown to coalesce into a discrete particle that is maintained at the bud tip. Mutations in five genes, **SHE1 <S000000027>**-**SHE5 <S000005215>**, cause defects in particle formation and/or localization of the ASH1 transcript. Factors at the destination of the mRNA transport remain to be identified.

Figure 3.8: Example of recognition and normalization of entities.
Two normalized mentions of gene and protein mentions are shown according to the identifiers of the Saccharomyces Genomics Database (SGD).

In the first figure, the mentions and their respective identifiers were annotated. This information is helpful because when used as an input to an entity normalization system, the exact location of the entities is usually needed in order to infer additional information from the context. The example in Figure 3.9 merely lists the identifiers of the genes and protein present in the text. However, it could be helpful as input to an information retrieval system, in order to search for publications related to the referred entities.

There are many difficulties in the biological normalization task, and in particular for the genes and proteins normalization. Variability and ambiguity are two of the major problems, as several synonyms (or aliases) might exist for a particular entity. Additionally, these various names may be written in different ways by distinct authors, as there is no standardization in the nomenclature for most of the organisms. For example, a given synonym may appear in uppercase or lowercase, or with the use of spaces or hyphens in parts of the name, as for example between the letters and numbers in its composition. The names may also vary due to mistakes in the spelling, which may arise some difficulties when trying to normalize the mention to its identifier.

Eukaryotic cells localize selected mRNAs to a region of the cell as a means to sequester proteins. Signals within the 3' untranslated region (3' UTR) facilitate mRNA localization by both actin and microtubule cytoskeletal systems. Recently, an mRNA in the yeast Saccharomyces cerevisiae, ASH1, was shown to coalesce into a discrete particle that is maintained at the bud tip. Mutations in five genes, SHE1-SHE5, cause defects in particle formation and/or localization of the ASH1 transcript. Factors at the destination of the mRNA transport remain to be identified.

S000001668 **S000000027** **S000005215**

Figure 3.9: Example of normalization of entities on document-level.
Normalization of gene and protein mentions are shown according to the identifiers of the Saccharomyces Genomics Database (SGD) without the recognition of the mentions in text.

The extensive use of acronyms is also a serious problem because sometimes a synonym can be referred with its full term or by its abbreviation. Also, the letters that compose an acronym not always correspond to the corresponding words of the full term. The acronym may include some extra letter, or just the opposite, it may miss some of the words of the full term. In addition, a particular abbreviation found in the text might not necessarily refer to the surrounding biological entities; it can refer to other processes or entities that are not even related to the molecular biology domain. Usually, the abbreviation and the full term are cited together only once in a publication.

Ambiguity is also an important issue in the entity normalization task, as a particular synonym might refer to different entities of the same organism or even of different organisms. The decision of the species to which the synonym refers is usually resolved with the use of context information. Finally, synonyms may match with common words, such as English words, which can end up being undetected by the system, when a stopword list is used.

Many solutions have been proposed for the gene/protein normalization task and most of them share the same sequence of steps: (a) extracting the mentions from the text; (b) a matching between the mention and a pre-processed dictionary of synonyms, one for each of the organism involved. Also, an optional last step includes filtering the results and/or performing a disambiguation among the candidates' identifiers, in case that more than one are found for a same mention.

The first step, the extraction of the gene and protein mentions, is usually performed by the same system responsible for the normalization (Fundel, Guttler et al. 2005), but sometimes it is carried out by one or more of the freely available systems, such as Abner (Settles 2005) or Banner (Leaman and Gonzalez 2008) taggers.

The second step, the normalization task, is highly dependent on the organism under study. For example, the nomenclature of genes and proteins for the *Saccharomyces cerevisiae* (yeast) is usually relatively simple while the nomenclature of the *Drosophila melanogaster* (fruit fly) sometimes matches with some English words.

Therefore, different organisms might require different strategies (Crim, McDonald et al. 2005) or specific curated dictionaries (Fundel, Guttler et al. 2005; Hanisch, Fundel et al. 2005), depending on the complexity of their nomenclature and the degree of ambiguity in the assigned synonyms. This is a problem because a name may or may not refer to distinct entities of the same species.

The gene/protein normalization task has received much attention from the scientific community in the last years due to the BioCreative challenges (Hirschman, Colosimo et al. 2005; Morgan, Lu et al. 2008). Stand-alone systems, such as GNAT (Hakenberg, Plake et al. 2008) and web-based ones, such as Whatizit (Rebholz-Schuhmann, Arregui et al. 2008) are available for performing normalization tasks.

3.4.2 Dictionary of synonyms

The construction of a dictionary of synonyms for the entities is an essential step in the normalization of biological entities. The dictionary will include a mapping of the identifiers to the various synonyms, according to a predefined database or ontology. These identifiers are the essence for the normalization of the entities. Two are the strategies usually performed:

- the use of a poor dictionary combined with an approximate matching between the mentions extracted from the text and the synonyms;

- investing more effort in expanding the dictionary with variations of the synonyms and the subsequent use of an exact (or less approximate) matching between the synonyms and the mentions extracted from the text.

From the initial list of synonyms of a determined organism, there are several operations that can be carried out in order to add variations to the synonyms or in order to remove those synonyms which are less helpful for the normalization task. The initial list of synonyms is usually automatically obtained from a database or ontology. It may also be built by joining more than one database or ontology (Liu, Wu et al. 2004). In the case of the gene/protein normalization, there are some dictionaries of synonyms available for use as base list, such as the ones provided by the BioThesaurus (cf. A.2) or by the first two versions of the BioCreative challenge (cf. B.2 and B.3).

The lists of synonyms provided by the BioCreative Task 1B (cf. B.2) were used as starting point for yeast and mouse in the works of (Fundel, Guttler et al. 2005) and in the ProMiner system (Hanisch, Fundel et al. 2005), and for three organisms (yeast, mouse and fly) in (Crim, McDonald et al. 2005). Alternatively, the dictionary of synonyms may be built using the information from one or more databases (Krauthammer, Rzhetsky et al. 2000; Fundel, Guttler et al. 2005), such

as the works of (Jenssen, Laegreid et al. 2001; Koike and Takagi 2004) and the systems BioTagger (Liu, Wu et al. 2004) and TextDetective (Tamames 2005). The list of synonyms may also be built based on curated documents, such as the work of (Tsuruoka and Tsujii 2003) which uses the GENIA corpus (Ohta, Tateishi et al. 2002). In this case, the list is composed exclusively of synonyms present in this set of documents, which can result in a very limited dictionary of synonyms. The initial list of synonyms may have some synonyms changed, added or removed through some operations of flexibility, expansion and exclusion.

Regarding the operations for flexibility, synonyms may be converted to lowercase and punctuation and special characters may be ignored and substituted for spaces, as well as special characters, for example, "03/03/1914" can be changed to "14 3 3". A synonym can be separated into parts, according to letters and numbers, such as the word "M5R" that may be converted to "m+5+r". Finally, the tokens which compose a synonym can be ordered alphabetically, for example, "cholinergic receptor, muscarinic 5" would be converted to "5 cholinergic muscarinic receptor".

New synonyms can be added to the initial dictionary through the operation of expansion. Plural may be derived from the original synonyms and Greek letter may be expanded in the subtypes, as for instance, "a" for "alpha" (Lau and Johnson 2007). Variations of the original synonyms can be automatically generated, especially when it is composed by a mix of letters, number and symbols (usually hyphens), such as the variations "Igf 1", "IGF-1" and "Igf1" (Schuemie, Jelier et al. 2007). Conversion of upper cases to lower cases, or vice-versa, can also be considered, especially for short synonyms. The expansion of long names to short names, or vice-versa, has been considered by many authors, usually using the algorithm developed by (Schwartz and Hearst 2003) for automatic acronym expansion. (Koike and Takagi 2004) have also proposed a methodology for the expansion of long names to acronyms and vice-versa. Some expansion operation can be specific to a determined organism, depending on its nomenclature. For example, the letter "h" can be added to the beginning of the human gene/protein synonyms (Cohen 2007).

Finally, some operations are carried out in order to remove less significant synonyms or to reduce the size of dictionary, and consequently, improve the time performance of normalization task. The exclusion of synonyms from the dictionary are usually carried out automatically, although some authors have developed some manual rules for this task, such as the work of (Fundel, Guttler et al. 2005) and the ProMiner system (Hanisch, Fundel et al. 2005). A stemmer (cf. 3.1.5) can be used in order to substitute the words by their stem. Synonyms which coincide with English words or stopwords may also be removed. Also very common is to exclude from the dictionaries those synonyms composed exclusively of numbers or whose length is less than two or three characters.

More specific to the biomedical domain, some authors (Schuemie, Jelier et al. 2007) have decided to remove those synonyms which stand for gene or protein families, usually represented by a set or words followed by a Greek letter or an Arabic or Roman number, such as "zinc finger protein 51". Some authors have developed some very specific rules when removing the synonyms, such as the exclusion of those synonyms which start with the token "LOC" or are preceded by "similar to" (Baumgartner, Johnson et al. 2007). In the work of (Crim, McDonald et al. 2005), synonyms are removed according to their degree of information, which is measured based on their presence in a set of training documents. This strategy calculates the conditional probability of a determined gene to be present in a document based on the occurrence of its synonyms in the training dataset.

3.4.3 Matching of synonyms

The matching procedure may be an exact or rule-based approximated one against an extensive curated dictionary of synonyms or an approximate matching. An example of the first approach is the work of (Fundel, Guttler et al. 2005) which uses an exact matching over a manually curate dictionary. The ProMiner system (Hanisch, Fundel et al. 2005) classifies the synonyms according to some predefined classes and also using a manually curated dictionary.

Regarding the approximate matching, the work of (Crim, McDonald et al. 2005) uses a string similarity approach based on the Jaro-Winkler metric (Cohen, Ravikumar et al. 2003). The TextDetective system (Tamames 2005) defines some manual rules in order to carry out a flexible comparison according to some predefined classes. The work of (Jenssen, Laegreid et al. 2001) uses a case sensitive or insensitive comparison depending whether the synonym is an acronym or long name, respectively. The work of (Krauthammer, Rzhetsky et al. 2000) uses the BLAST algorithm to accomplish the comparison between the text and the synonyms. The text and the synonyms are converted to 4-letters alphabet of the amino acids and these are further compared using BLAST. The local alignments found constitute the mentions (in the text) and the selected synonym. Some authors have trained a machine learning algorithm in order to decide if the matching is correct or not, such as the logistic regression algorithm in (Tsuruoka, McNaught et al. 2007) which use bigrams, prefixes and suffixes as features. The work of (Wermter, Tomanek et al. 2009) uses a similar strategy with some extra features such as the molecular weight, Greek letters or the gene name specifier.

3.4.4 Post-processing

A post-processing step is usually carried out with the candidates obtained in the matching step. It usually consists of filtering out the false positives and performing disambiguation among the candidates.

The filtering step removes those synonyms which were incorrectly matched. For example, the work of (Crim, McDonald et al. 2005) uses a maximum entropy classifier previously trained with positive and negative examples from training documents. The classifier decides whether a matching between a certain mention and synonym is correct or not. The work of (Fundel, Guttler et al. 2005) uses a post-processing step which decides whether a match is valid or not based on some keywords, such as "cells" and "domain", which usually represent other types of entities instead of genes and proteins. In (Jenssen, Laegreid et al. 2001), some training documents are analyzed in order to search for the presence of genes in the title or in the abstracts and some manual rules were developed for the filtering decision.

The disambiguation step is the most common of the post-processing step. It resolves the correct identifier associated to a determined mention when more than one identifier has been matched to it. Different approaches have been proposed for the disambiguation of the mentions, such as context information (Hakenberg, Plake et al. 2008), machine learning based filters, as in the ProMiner system (Hanisch, Fundel et al. 2005), or a similarity measure between the abstracts and the disambiguation vectors for each gene (Liu, Wu et al. 2004). In the work of (Farkas 2008) the influence of the co-authors of the publication in the nomenclature of the entities is used for disambiguating the mentions. Sometimes the same strategy may be used for the disambiguation and the false positive filtering, such as the so-called "semantic similarity score" in the work of (Wermter, Tomanek et al. 2009).

3.5 Evaluation Metrics

The evaluation of the results for all the tasks under consideration in this thesis is carried out using the concepts of precision, recall and F-Measure (Shatkay and Feldman 2003). Details on these metrics are presented in this section.

The first step for calculating these measures is counting the number of correct and incorrect results that have been obtained. The correct answer will be given by a curated gold-standard corpus, usually annotated manually by experts in the field, such as in the work of (Krallinger, Morgan et al. 2008). Therefore, based on the correct results provided by a determined benchmark corpus for a given task, the following values are calculated and shown in Figure 3.10.

- True Positives (TP): the number of correct answers performed by the system, i.e., those results which also appear the annotated corpus;

- False Positives (FP): the number of incorrect answers performed by the system, i.e., those results which do not appear in the annotated corpus;

- False Negative (FN): the number of answers which are present in the annotated corpus but that were not found by the system.

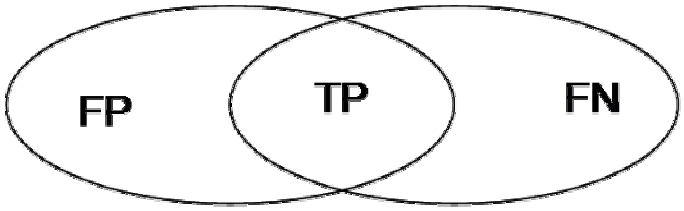


Figure 3.10: Venn diagram for the evaluation of the results.
The left ellipse represents the results returned by the system and the right one the correct answers, usually given by a gold standard corpus. The intersection of both groups correspond the true positives.

Based on the three values above, we can define the metrics of precision, recall and F-Measure, as follows:

- Precision (P) (cf. Equation 3.1) is the fraction of correct answers among all those returned by the system. Thus, it is the ratio between the true positives and the sum of true positives and false positives, i.e., all the results that has been returned by the system:

$$P = \frac{TP}{TP + FP} \qquad \textbf{Equation 3.1}$$

- Recall (R) (cf. Equation 3.2) is the fraction of correct answers returned by the system which are effectively correct according to the gold corpus. Thus, it is the ratio between the true positives and the sum of true positives and false negatives, i.e., all the results of the official corpus:

$$R = \frac{TP}{TP + FN} \qquad \textbf{Equation 3.2}$$

- F-Measure (FM) (cf. Equation 3.3) is the harmonic mean between precision and recall:

$$F - Measure = \frac{2 \cdot P \cdot R}{P + R} \qquad \textbf{Equation 3.3}$$

Distinction should be made between the micro and macro-averaging results. The first one calculates the precision, recall and f-measure across all documents, i.e., they are based on the sum of the true positives, false positives and false negatives across all the documents under consideration. It gives a general view of the performance of the system for the whole corpus.

In contrast, the macro-averaging calculates the same concepts separately for each document of the corpus, and then performs an average of these values across all documents. Thus, it helps deciding which system performs reasonably well for each document, instead of the corpus as a whole. The micro-averaging f-measure is usually the standard measure used for comparison among systems. We refer o this one throughout the thesis when F-Measure (FM) is discussed.

3.6 Summary of the chapter

In this chapter we have presented a serie of tasks that are actively used and studied in biomedical text mining. Even if these problems and techniques seem to be unconnected, they are usually properly combined to create an analysis workflow to solve text mining problems.

We started this chapter by presenting in section 3.1 an introduction to natural language processing. We have described in details tools which are usually used for shallow linguistics tasks, such as such as sentence splitters, tokenizers, part-of-speech, stemmer, lemmatizer and chunkers, as well as the deep syntactic parsers.

The named entity recognition task is presented in section 3.2, and more especially for the extraction of genes and proteins entities from the scientific literature. The approaches which have been proposed for this popular task are described, which are usually based in machine learning algorithms or manual developed rules. For the representation of the tokens of the texts, orthographic and morphological features are usually used.

The extraction of biomedical relationships (cf. 3.3) is an important task in order to relate different entities of the same type, such as the protein-protein interaction, or from different classes, such as the extraction of biomedical events (gene expression, regulation, etc.). The methodologies that have been proposed for these heterogeneous and complex tasks range from co-occurrence, pattern matching, manual rules and linguistic methods based on the heavy use of parsers to a variety of machine learning algorithms.

Finally, the next step following the extraction of the entities from a text is their normalization according to some predefined database or ontology, as has been described in section 3.4 for the genes and proteins domain. The approaches for this problem are usually based on dictionary lookup and machine learning algorithms. A post-processing step is usually needed in order to filter false positives and for the disambiguation of the identifiers in those cases in which more than one entity is assigned to the same mention.

In section 3.5, the metrics used for the evaluation of the methodologies proposed in this thesis have been described, which are precision, recall and f-measure. We also describe the building blocks for these measures (true positives, false positives and false negatives), as well as the distinction between micro-averaging and macro-averaging.

CHAPTER 4 EXTRACTION OF BIOMEDICAL ENTITIES AND RELATIONSHIPS

This section describes the methodologies proposed for the extraction of biomedical entities and relationships. The use of case-based reasoning, which is introduced in section 4.1, is proposed for these tasks. Two approaches using case-based reasoning are presented here, one developed specifically for the extraction of genes and protein mentions (cf. 4.6) and a more general one (cf. 4.2) which has been used for the extraction of event triggers and biomedical events.

4.1 Case-Based Reasoning

Case-based reasoning (CBR) (Aamodt and Plaza 1994) is an artificial intelligence approach and a sub-field of machine learning. It consists of using specific knowledge from past examples (cases) to resolve a new problem. It is carried out by looking for a similar past case and reusing it for the solution of the new problem. In other words, new solutions are inferred (or remembered) by using the solution of the past cases. A case can be defined as a past situation which has been appropriately saved in order to be able to be reused to solve future problems. A new case is thus a new problem waiting to be solved. CBR is considered as a sustained learning as new solved cases may be retained in order to be used for future problems. New cases can also be saved when a solution for a certain situation has been successfully solved. Additionally, the reason of the failure might also be retained by the system in order to avoid the same error to happen again. Figure 4.1 shows an interesting example of the way a new problem can be solved by using the solution used from a past case.

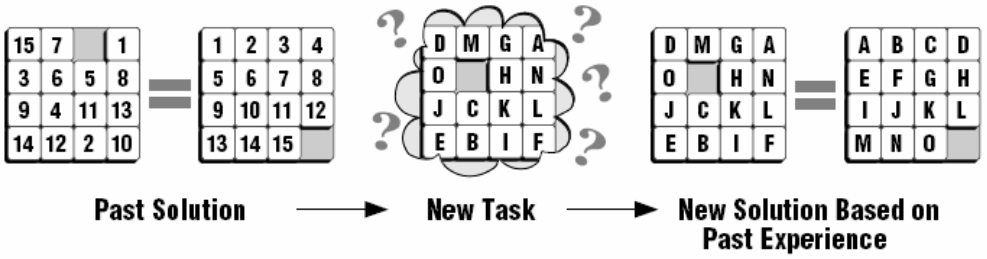


Figure 4.1: Example of case-based reasoning.
In this example, the solution used for solving a puzzle composed of number might also be used for solving a puzzle composed of letter instead. Figure extracted from (Slade 1991).

One of the advantages of CBR over other machine learning methods is that it usually performs no generalization of the solution. Instead, specific information of the past situation is used for the solution of the new ones, which is usually easier than generalizing. However, CBR may also represent generalizations, as cases may represent a single situation or a set of similar ones. In order to be effective, a CBR system must be able to effectively represent the past situation and integrate each case in a knowledge base as well as be able to retrieve a similar case in an appropriate time period.

Another advantage of the CBR is the possibility, by means of checking the features that compose the case-solution, of getting an explanation of why a certain category has been assigned to a given token. In addition, the base of cases can be used as a natural source of knowledge from which to learn extra information about the training dataset.

Case-based reasoning is especially appropriate when an explanation is needed along with the solution, i.e., a description of the reason of the solution which has been proposed. The retrieved case can therefore be used as an example of what have been done for a similar past situation. Two typical examples occur in the medicine and the financial domains. The diagnosis and treatment used for a previous patient may be helpful to give a solution for a new one with similar symptoms. Likewise, the decision of giving or not credit to a client by a bank may be influenced by past situations.

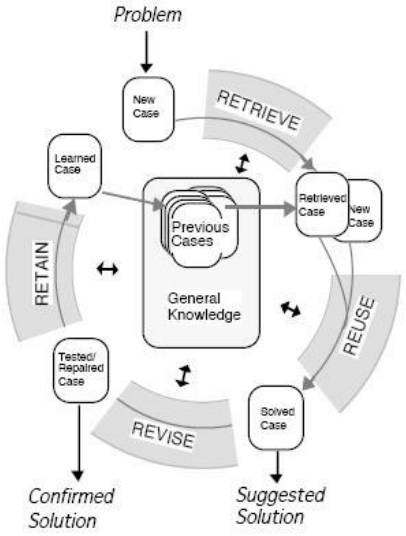


Figure 4.2: Cycle of the case-based reasoning.
The "four Rs" which compose a case-based reasoning cycle are shown: retrieve (a new case for solving a problem), reuse (of the retrieved case to solve a problem), revise (the proposed solution when not correct) and retain (a new case when necessary). Figure extracted from (Aamodt and Plaza 1994).

Case-based reasoning can be used for a variety of tasks (Kolodner 1992) such as: designing a solution for a problem which is defined by a set of constraints, planning a process (set of steps) in order to obtain a certain result, giving an explanation to some situations, making persuasive arguments to convince others or interpreting a new situation. CBR may also be used for diagnosis, which can also be viewed as a classification problem in which some solutions are proposed according to a given set of symptoms. This is the case of the solutions we propose for text mining in this thesis, in which the symptoms are features of the text and the solution might be its classification as a gene or not (cf. 4.6), for instance.

The CBR cycle (cf. Figure 4.2) may consist of four steps, the so-called "four Rs": retrieve, reuse, revise and retain. The cases are saved in a knowledge base and can be *retrieved* to be *reused*, when a solution is needed for a new case. The solution may need to be *revised* in order to fit the new case and if correct, the new case can be *retained* for future use.

The retrieval of a case starts with a description of the problem and by identifying the best features to be considered. It usually consists of first matching the most similar cases and then searching and selecting the best of them, i.e., the more appropriate ones according to the solution. The result of this process is the most similar cases which are going to be reused (and maybe revised) for the solution.

For the reuse, differences between the new case and the retrieved one are taken into account in order to decide which parts of them are useful for the new case. Sometimes, the adaptation of the solution is needed to fit the new case. The revision takes part when the solution proposed by the case was not appropriate for solving the new case. The retrieved case might then be changed in order to learn from the failure. In case of success, the case may be retained in the knowledge base for future use, especially when an adaptation of the solution was necessary. In the methodologies proposed here, no revision of the solution and no retainment of a new case are performed, as will be further discussed (cf. Chapter 6).

In CBR systems, a knowledge base (or a case memory) needs to be developed in order to allow searching and matching of cases. Many methods have been proposed for integrating a new case into the memory. The construction of a memory base usually include the following tasks: finding an appropriate structure to define the contents of the case and organizing it and indexing it for an effective retrieval, reuse and retainment.

In CBR, some general knowledge may be used to support the above four steps, although it is not mandatory. For instance, for the domain of the gene/protein recognition, a dictionary of synonyms for genes and proteins could be used. In contrast to the general knowledge, the knowledge base where the cases are saved represents the specific knowledge. In our CBR methodologies proposed in this thesis, few or no general knowledge has been employed.

As far as we know, CBR (or similar approaches) have not been widely used in the biomedical text mining domain, but some previous works have been reported. CBR has been previously used for the biomedical term classification in the MaSTerClass system (Spasic, Ananiadou et al. 2005). A memory-based approach (Morante, Van Asch et al. 2009) has been proposed for the extraction of biological events, the same task for which we developed some methodologies (cf. 4.3 and 4.4) However,

CBR has been widely used in others text mining domains, as presented in (Weber, Ashley et al. 2005).

4.2 General Methodology

Our general methodology proposes the use of case-based reasoning for the extraction of biomedical entities and their relationships. The procedure described here has been evaluated for the extraction of biological event triggers (cf. 4.3), a named entity task, and for the extraction of relationships, namely: the extraction of biological events (cf. 4.4) and associations between diseases and treatments (cf. 4.5). The general methodology is common for all these tasks, only that some particularities have been implemented for each of them, especially the features that may vary depending on the types of entities or the relationships under consideration.

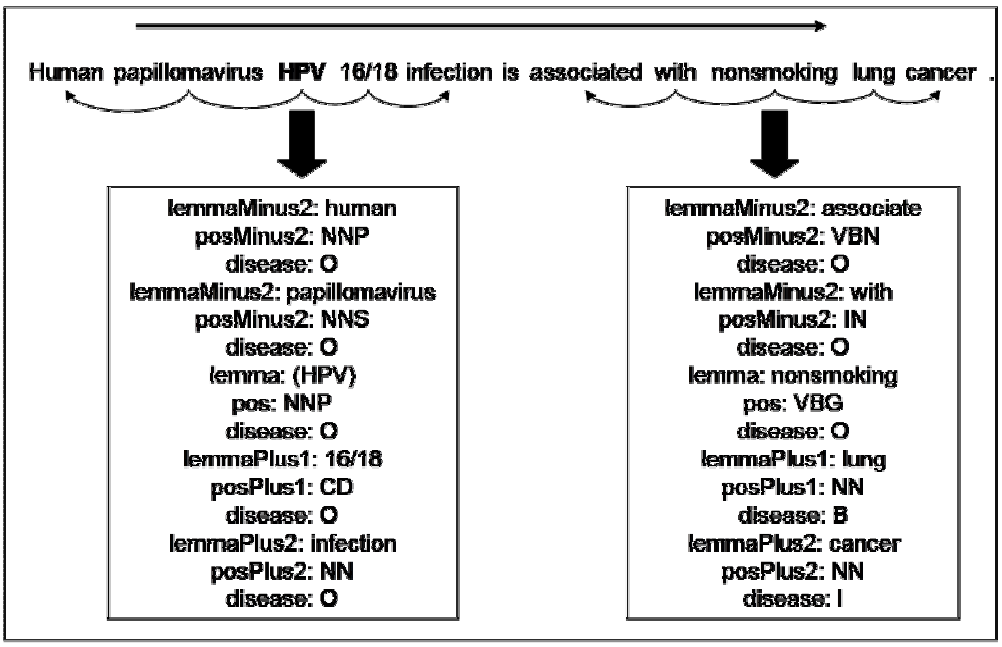


Figure 4.3: Creation of cases for the recognition of diseases.
Two examples of cases are shown, each one for a window of token of size two (two preceding and succeeding tokens). Each token is represented by three features: the lemma, the part-of-speech tag and whether it is a disease of not. The disease tag uses the BIO format. The arrow on the top indicated the direction in which the text is being read.

The general methodology consists of training and testing steps of the case-based reasoning algorithm. In the training step, several cases are stored in one or more bases of cases. The input data (usually the abstract of a document) is represented as a set of cases which are composed of some predefined features. In information extraction, a case usually represents part of the text, and not the complete document, as in text classification, for instance. Therefore, a case can correspond to a window of token of a particular size, for instance, a token and the five words which come before and after it.

When creating the cases, the text of a document can be read in the forward or backward direction during both training and testing steps, and even as a combination of both of them. An illustration of the training step is presented in Figure 4.3. This process is repeated for each of token in the text, and each case in saved into the base of cases, without repetition of the same value for the features. Instead, each case has an attribute that correspond to its frequency in the training dataset. One or more features may be defined as unknown during the testing step, the ones whose solution will be given by the cases extracted from the base. For the example shown in Figure 4.3, the unknown feature in the testing phase would be the "disease" tag, i.e., whether the token is a disease or not.

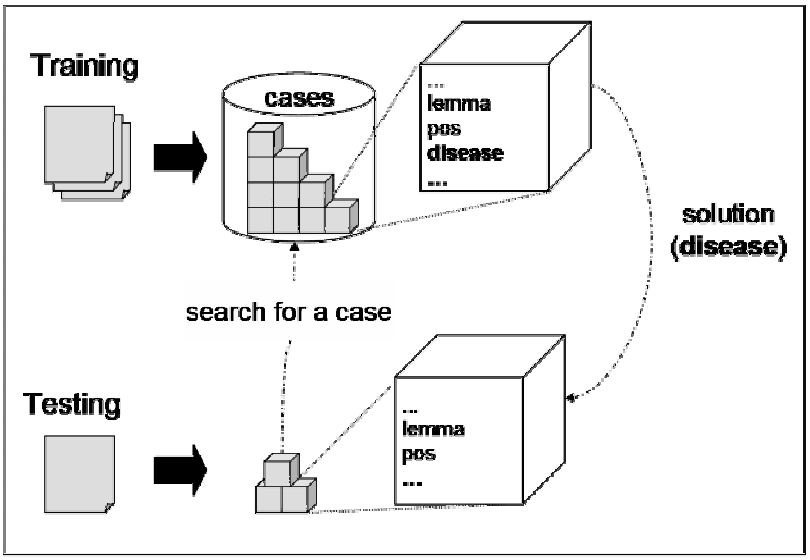


Figure 4.4: Training and testing steps for the case-based reasoning methodology.
In the training step, the document is converted to a set of cases which are saved in a base of cases. In the testing step, the document is converted to cases using the same set of features and a search in the base of cases is carried out in order to find the cases most similar to them. Here, the "disease" feature is missing in the testing case, whose value will be given by the selected case from the training step.

During the testing step, the same representation of cases is used for the input data, i.e., by considering the same features used in the training step, except the ones that are configured as unknown. The system then searches the bases for the cases the most similar to these new cases. More than one case might be returned for a given case-problem and the final solution to the problem may be obtained, for example, based on a voting scheme. The final solution is given by the assignment to the unknown features the value of the corresponding features of the cases which have been selected. Figure 4.4 show as overview of the training and testing steps.

4.2.1 Representation of the cases

In our methodology, the cases are represented by a context, which represents a sequence of consecutive tokens in a sentence. In Figure 4.3, two contexts were shown, each one composed of five tokens: “Human papilomavirus HPV 16/18 infection” and “associated with nonsmoking lung cancer”.

Being a case, and in order to be used in our case-based methodology, a context has to be composed by features. These features can be related to the tokens in their composition or to the whole context itself. For example, a feature which represents the lemma of the token usually has a different value for each of the tokens of the context. On the other hand, the type of named-entity, e.g., whether it is a gene or not, is a feature related to the whole context, i.e., the limited group of tokens.

The boundaries which set the limits of the context, i.e., the set of tokens, can be defined in two ways in our methodology, namely: as a window of tokens of a predefined length (limited by some predefined start and end tokens) or as structure of variable size based on some given named-entities. These types of contexts are defined below.

4.2.1.1 Context based on a window of tokens

The window of token is a structure which represents a local context of the sentence. It is composed of a pre-specified number of tokens. Therefore, it is limited by predefined start and end tokens. In our methodology, the window is always composed of a base token and the “m” (minus) preceding tokens and the “p” (plus) following tokens. The window is therefore represented with the notation [-m,+p] and the length is given by (|m| + |p| +1). The window may be symmetric or not, i.e., the values assigned for “m” and “p” may be different. Figure 4.5 presents an example of a case represented as a window of tokens.

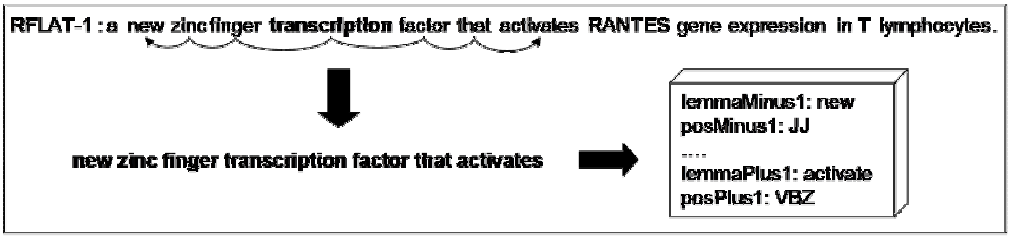


Figure 4.5: Example of a case represented as a window of tokens.
The case is represented as a context [-3,+3] which includes the base token (“transcription”), the preceding three tokens (“new zinc finger”) and following three tokens (“factor that activates”). The values of the lemma and part-of-speech tag features are shown for the first (“new”) and last (“activates”) tokens.

For the window of tokens, we define features related to the whole structure, such as being a gene or not, or related to each token exclusively, such as the stem or lemma (cf. 3.1.5), the part-of-speech tag (cf. 3.1.4) and the chunk tag (cf. 3.1.6). A

feature such as a dependency tag (cf. 3.1.6) is not represented in such a structure. This window of tokens is more suitable for the named entity recognition task and has been used for the extraction of the biological event triggers (cf. 4.3) and for the extraction of gene and proteins (cf. 4.6). It should be noted that the wider the window of tokens, the higher is the number of distinct cases to be inserted into the base of cases. It would then take more time to search the base for the similar cases during the testing step, as well as inserting new cases during the training and retaining steps (cf. 4.1).

4.2.1.2 Context limited by predefined entities

Here the context is represented by a variable number of consecutive tokens in a sentence. The boundaries are usually defined by two or more predefined tokens or given entities, such as a disease and a treatment, in case of interactions between these types of entities (cf. 4.5). However, it can also de defined according to the syntactic tree of the sentence, for instance, all the tokens which compose a certain noun phrase.

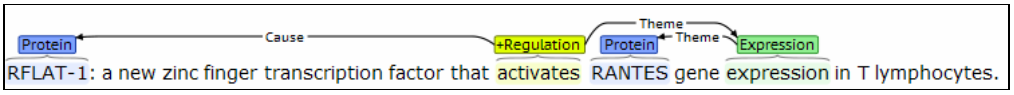


Figure 4.6: Example of a biological event.
The proteins are shown in blue, the trigger of the positive regulation event in yellow and the trigger of the expression event in green. Example extracted from the document 10023774 of the BioNL Event Extraction corpus available in Stav visualization tool[10].

As an example, we propose the sentence shown in Figure 4.6 which contains two proteins, which were given in the BioNLP Event Extraction corpus (cf. B.4) and two event triggers, which could have been extracted in a previous step (cf. 4.3). When interested in deciding the protein related to each of the trigger events we could define four contexts here, using pairs composed of one protein and one trigger event: "**RFLAT-1**: a new zinc finger transcription factor that activates RANTES gene expression"; "**RFLAT-1**: a new zinc finger transcription factor that activates" (C2); "activates **RANTES**" (C3) and "**RANTES** gene expression" (C4). The limits of each context are one protein (in bold) and one event trigger (underlined). We would not, for instance, consider a context using two proteins as limits. However, this approach would be suitable for extracting protein-protein interactions.

From Figure 4.6, we can learn that the contexts C2 and C4 are correct, as there is a direct relationship between protein "RFLAT-1" and the trigger "activates", as well as a direct relationship between the protein "RANTES" and the trigger "expression". For similar reasons, the contexts "C1" and "C3" are wrongs. Being a training document, these wrong contexts (cases) would be used as negative

[10] http://corpora.informatik.hu-berlin.de/index.xhtml#/bionlp2009st/training/10023774

examples. Alternatively, the contexts could be defined not only by the entities (proteins and triggers), but also by some a predefined number of tokens that precedes or succeeds the boundary entities. After defining the limits of the context, it is represented in the case according to some pre-defined features, such as the example shown in Figure 4.7.

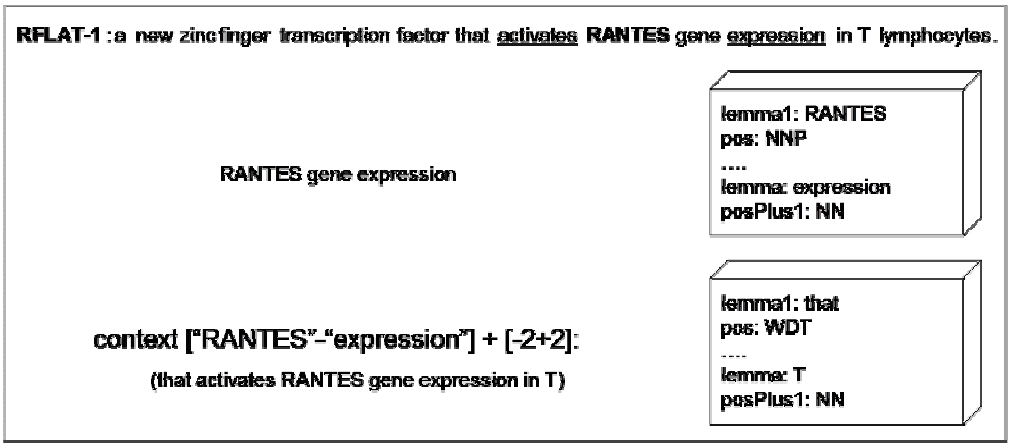


Figure 4.7: Examples of cases which represent context defined by given entities.
The case is represented by the part of the sentence limited by two entities, the protein "RANTES" and the trigger event "expression". The second example also includes the preceding ("that activates") and following ("in T") two tokens. The values of the lemma and part-of-speech tag features are shown for the first and last tokens of the context.

The disadvantage of using named-entities for defining the limits of the context is that the latter might not contain the whole syntactic sub-structure, for instance, the complete noun or verb phrase. For example, for the contexts defined above and based on the syntactic structure of the sentence showed in Figure 4.8, context "C4" ("RANTES gene expression") represents a complete noun phrase. However, context "C3" (activates RANTES) is incomplete as it fails to include part of the noun phrase ("gene expression") and the prepositional phrase ("in T lymphocytes"). Alternatively, one could consider the minimal syntactic sub-structure which includes certain entities. Thus, the context "C3" would contain the text "activates RANTES gene expression". However, the use of deep parsers (cf. 3.1.6) was out of the scope of this thesis.

The contexts discussed here is a type of representation which is suitable for the extraction of relationships. Depending on the task, the entities which define the boundaries may be two named entities, such as two proteins for the protein-protein interaction task, or a protein and an event trigger, for the biomedical event extraction. In this thesis, this type of context has been used for the extraction of the arguments of the biological events (cf. 4.4.2) and for the extraction of relationship between diseases and treatments (cf. 4.5).

```
(ROOT
  (UCP
    (ADJP (JJ RFLAT-1))
    (: :)
    (NP
      (NP (DT a) (JJ new) (NN zinc) (NN finger) (NN transcription) (NN factor))
      (SBAR
        (WHNP (WDT that))
        (S
          (VP (VBZ activates)
            (NP (NNP RANTES) (NN gene) (NN expression))))))))
```

Figure 4.8: Syntactic tree for the biological event example.
The sentence was parsed using the online version of the Stanford parser[11].

4.2.2 Automatic generation of contexts

In our methodology, the contexts are automatically generated based on the named entities which have been previously identified and on the task under analysis. For example, for the BioText corpus (cf. B.5), only two entities are involved, a disease and a treatment. However, the biological events involved in the BioNLP'09 Event Extraction task (cf. B.4) are much more complex as there many types of entities (triggers, proteins, sites, locations, other events, etc.), and an event might be composed of many arguments, some of them optional.

Therefore, the first step in generating the candidates for contexts is to identify the entities in the text and separate them in "bags of entities" by sentence, as shown in Figure 4.9. Given the bags of entities, the contexts are automatically generated by combining one or more entities from each of the bags, according to the type of relationship under consideration. The length of a context depends on the representation of the context being used (cf. 4.2.1).

Figure 4.10 shows an example of the contexts automatically generated for one of the sentences shown in Figure 4.9. Two bags of entities were created, one for the treatments, which contained the entity "Metylphenidate", and one for the diseases, which contains the entities "epilepsy" and "ADHD". A relationship between a disease and a treatment is composed of only two entities, one of each type. Consequently, two contexts could be generated from these bags of entities, e.g., one from "Metylphenidate" to "epilepsy" and one from "Metylphenidate" to "ADHD". The context could be longer and also include the surroundings tokens, as also shown in Figure 4.10, depending on the size of the window being used (cf. 4.2.1.1). Depending on the complexity of the relationships, the generation of the context may be a much harder task, as will be described in section 4.4.2 for the event extraction task.

[11] http://nlp.stanford.edu:8080/parser/index.jsp

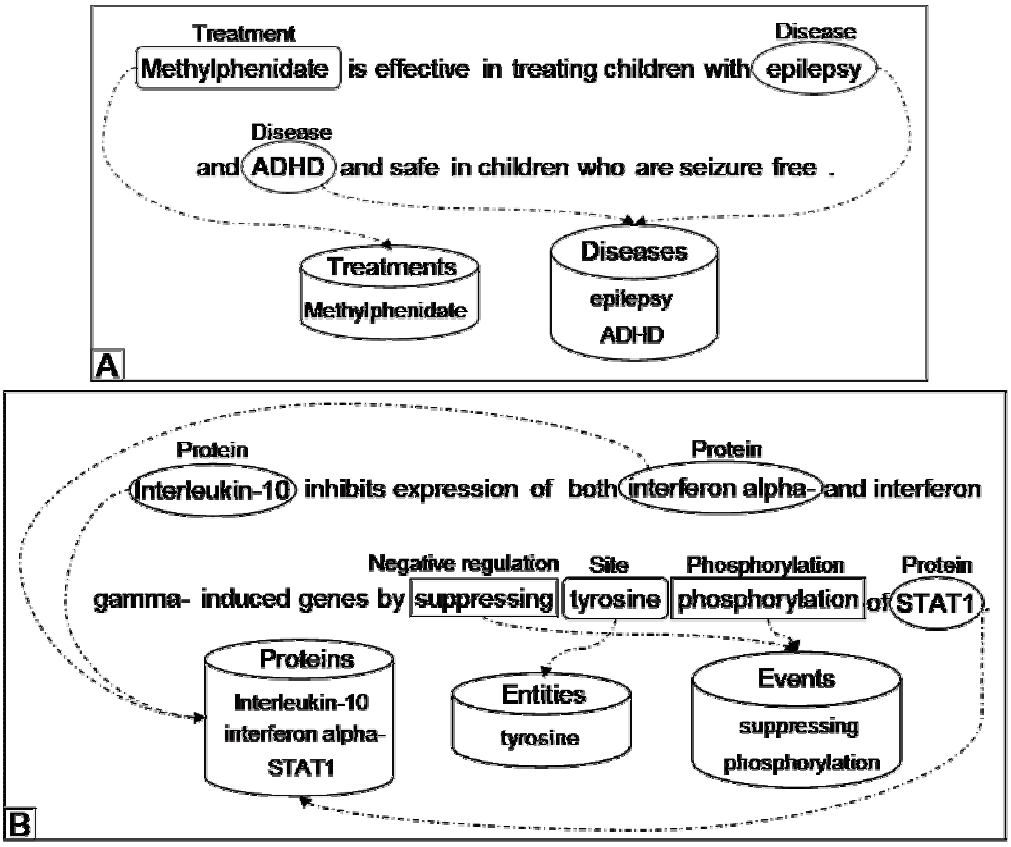


Figure 4.9: Examples of the bags of entities generated for a sentence.
Examples are shown for the BioText corpus (A) and for the BioNLP'09 Event Extraction corpus (B). In A, diseases are represented as ellipses and treatments are represented as rounded rectangles. In B, Proteins are represented as ellipses, event triggers as rectangles and other entities (sites) as rounded rectangles. One bag of entity is available for each type of entity.

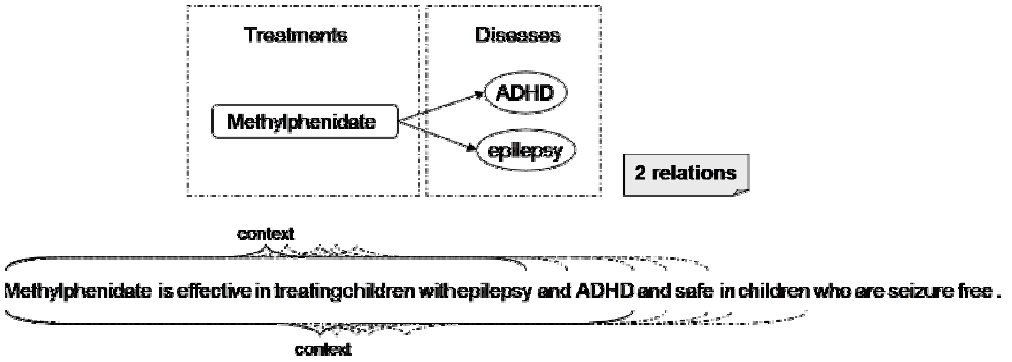


Figure 4.10: Automatic generation of contexts for the disease-treatment relationship.
Two bags of entities are shown, one for the disease and one for the treatments. Two contexts are generated, using one entity from each bag of entity. The length of the context may be delimited by the entities only (plain line) or by setting the number of tokens that come before and after them (dashed lines).

After the generation of the contexts, they are converted into cases, which are then inserted into the base of cases. This procedure is similar for the training and testing steps. The next section will describe the methods used for retrieving a case from the base.

4.2.3 Searching the base for a case

During the testing step, cases are retrieved from the base of cases in order to be used as a solution. As input, the cases which represent the contexts of the testing text are used, according to the chosen representation (cf. 4.2.1). Here the search for a case is carried out in two ways, using a binary search for the cases which exactly match as many features as possible with the input case, or by performing a global alignment between some of the cases in the base with the input case. Both methods are described in details below. More than one case might be returned for a given input case. The final solution would be then obtained on the basis of a voting scheme.

4.2.3.1 Case similarity based on exactly matching features

This strategy consists on looking for a case with exactly the same values for the features of the case used as input. Depending on the task, some features may be defined as mandatory or optional. When searching for the most similar case, the system tries to find cases with exactly the same values for the mandatory features and matching as many optional features as possible. More than one case may satisfy this condition and we consider as solutions all those cases which match the maximum number of optional features, no matter which features have been matched. In this thesis, this is the searching strategy which has been used for the extraction of the trigger, site and location argument parts of the biological events (cf. 4.3.1).

4.2.3.2 Similarity based on the global alignment of the features

This method consists in performing an initial search for cases which match the mandatory features of the input case and then calculating a global alignment between each retrieved case and the input case. Using a global alignment, the position of the tokens in the sentence is not that relevant, as long as their order in the sentence is similar. A high similarity is also achieved when tokens with similar features, such as part-of-speech, appear in similar order in the text.

This methodology has previously been used as part of a CBR algorithm for biomedical term classification in the MaSTerClass system (Spasic, Ananiadou et al. 2005). By default, for any feature, the inclusion and exclusion costs are 1 (one) and the substitution cost is 0 (zero) for substitutions of equal features with equal values, and 1 (one) otherwise. However, sometimes, specific costs may need to be defined for certain features.

For the part-of-speech tag feature, we have used costs inspired by the ones used in the MaSTerClass system for term identification. The costs for the inclusion, exclusion or substitution are therefore based on the costs proposed in the Table 1 of the work of (Spasic and Ananiadou 2005), reproduced here in the thesis in Table D.1 (Appendix, page 182). We have mapped the tags presented in this work to the part-of-speech tags returned by the Stanford parser, as presented in Table 4.1.

Columns and rows in Spasic's cost matrix	**Meaning**	**Stanford parser POS tags**
term	Terms, nouns	NN,NNS,NNP,NNPS,CD,FW
aux	Modal verbs	MD
adj	Adjectives	JJ,JJR,JJS
adv	Adverbs	RB,RBR,RBS,RP
cnj	Conjunctions	CC,WRB
det	Determiners	DT,PDT
prep	Prepositions	IN,TO
pron	Pronouns	PRP,PRP$,EX,POS,WDT,WP,WP$
pun	Symbols, punctuations	LS,SYM,UH,HYPH and the puntuation marks $(),:.
v	Verbs	VB,VBD,VBG,VBN,VBP,VBZ

Table 4.1: Mapping of the MaSTerClass' chunk tags to the Stanford POS tags.
The mapping of the tags in the matrix cost of the MaSTerClass system to the part-of-speech tags is presented, in order to allow the same costs to be used in global alignment of this feature.

We consider as solution those cases whose global alignment score in relation to the input case is below a certain threshold, which was automatically defined for each input case using Equation 4.1 (Spasic, Ananiadou et al. 2005). In this equation, "min" is the minimum score; "average" is the average of all scores and "d" is a predefined parameter in the range 0 to 1. All experiments carried out in this work have used "d" as zero. Thus, the threshold is given by the minimum score.

$$t = \left[\min + \left(average - \min\right) * d\right] \qquad \textbf{Equation 4.1}$$

Having the cases which have been retrieved as solution for the input case, the values for the missing features are given by a majority voting among the features of the retrieved cases. The next sections will present the evaluation of the methodology presented in this section for some tasks, namely: recognition of biological events triggers (cf. 4.3), extraction of biological events (cf. 4.4), extraction of relationships between diseases and treatments (cf. 4.5) and recognition of genes and proteins (cf. 4.6).

4.3 Recognition of biological event triggers

This section and section 4.4 will describe the methodology we propose for the extraction of biological events using the BioNLP'09 Event Extraction corpus (cf. B.4). This corpus contains annotation for nine types of biological events: localization (LOC), binding (BIN), gene expression (EXP), transcription (TRA), protein catabolism (CAT), phosphorylation (PHO), regulation (REG), positive regulation (POS) and negative regulation (NEG). Due to shortness of space in tables, sometimes we will refer to the abbreviations instead of the long name of each event.

When the biomedical entities involved in a certain task are not given, a named entity extraction step is needed in order to extract them, as it is the case of the event triggers, sites and locations entities in the BioNLP'09 Event extraction task (Kim, Ohta et al. 2009). In this corpus, however, the proteins are already provided and did not need to be previously extracted. In this section we describe two methodologies for the extraction of the event triggers, the one that participated in the BioNLP'09 Event extraction challenge (Neves, Carazo et al. 2009), which is mainly based on shallow linguistic features (cf. 3.1.6), and an improvement of this methodology, which also uses some deep parsing features (cf. 3.1.6).

4.3.1 Recognition of trigger events based on shallow linguistic features

The approach used for the extraction of the event triggers is based on the general methodology proposed in section 4.2. Some specific procedures used only for this task will be described here. The features which compose a case during both the training and development steps are shown below, along with their short names:

- the token itself (token);
- the token in lower case (lowercase);
- the stem of the token (stem) (cf. 3.1.5);
- the shape of the token (shape);
- the part-of-speech tag (posTag) (cf. 3.1.4);
- the chunk tag (chunkTag) (cf. 3.1.6);
- the biomedical entity tag (entityTag);
- the type of the term (termType);
- the type of the event (eventType);
- and the part of the term in the event (eventPart).

The stem of a token was extracted using an available Java implementation[12] of the Porter algorithm (Porter 1980), while the "posTag", "chunkTag" and "entityTag"

[12] http://tartarus.org/~martin/PorterStemmer/

features were provided by the GENIA Tagger (Tsuruoka, Tateishi et al. 2005). The "shape" feature is given by a set of characters that represent its morphology: "a" for lower case letters, "A" for upper case letters, "1" for numbers, "g" for Greek letters, "p" for stopwords (cf. Appendix E.1) , "$" for identifying 3-letters prefixes or suffixes or any other symbol represented by itself. Here are some few examples for the shape feature: "Dorsal" would be represented by "Aa", "Bmp4" by "Aa1", "the" by "p", "cGKI(alpha)" by "aAAA(g)", "patterning" by "pat$a" ('$' symbol separating the 3-letters prefix) and "activity" by "a$vity" ('$' symbol separating the 4-letters suffix). No repetition is allowed in the case of the "a" symbol for the lower case letters.

The "termType", "eventType" and "eventPart" features are specific to the event detection task and were extracted from the annotation files (.a1 and .a2) which are part of the corpus. The "termType" feature is used to identify the type of the term in the event problem, and it is extracted from both annotation files .a1 and .a2, i.e. the ones which the identifiers starts with a "T". The "eventType" feature represents the event itself and it is extracted from the event lines of .a2 annotation file, i.e. the ones that starts with an "E". Finally, "eventPart" represents the role of the token in the event, such as entity, theme, cause, site or location. More details on the BioNLP'09 Event Extraction corpus are presented in section B.4.

The "termType", "eventType" and "eventPart" are the features which are unknown and whose values are to be given by the cases retrieved from the base. Figure 4.11 illustrates one example of these features for an extract of the annotation of the document "1315834" from the training dataset.

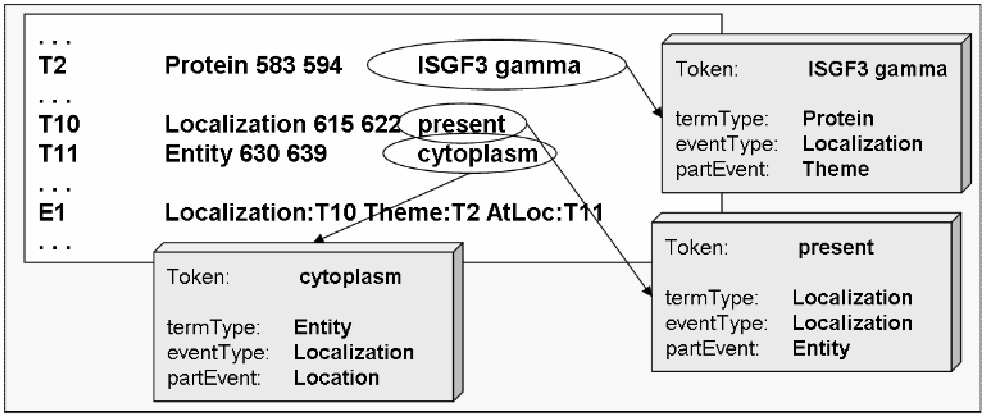


Figure 4.11: Example of values for the features related to an event.
The example present values for the features termType, eventType and partEvent for three tokens ("ISGF3 gamma", "present", "cytoplasm") which are arguments of a localization event.

Usually, one case corresponds for each token of the documents in the training dataset. However, more than one case may be created from a token, as well as none at all, depending on the predefined features. For example, some tokens may derive

in more than one case due to the shape feature, as for example, "patterning" ("pat$a", "a$ing", "a"). Also, according to the strategy used when saving the case into the base, some tokens may be associated to no case at all, for example, by defining a determined feature as mandatory. In this task, in order to reduce the number of cases to be saved, and consequently in order to reduce the retrieving time, only those tokens related to an event are retained, i.e., tokens with not null value for the "termType" feature.

For this task, we have considered the text in the forward direction exclusively. The cases here are are representation of the context as a window of token (cf. 4.2.1.1). Regarding the window of token, we use a (-1,0), i.e., for each token, we process the features of the token itself and of the preceding one, exclusively. Table 4.2 shows which of the features described above are used for the training and testing cases, a well as those which are unknown to the system in the testing step.

Many experiments have been carried out in order to decide which features to use for each token ("0" or "-1"). The higher the number of features under consideration, the greater is the number of cases to be saved and the higher is the time needed to insert and search for a case. Here relies therefore the importance of choosing a small an efficient set of features. For this reason, in order to reduce the number of cases, the "shape" features has not been considered for the preceding token (-1), as it usually results in more than one case per token.

Features / Tokens	**Training**		**Testing**	
	-1	**0**	**-1**	**0**
stem	✓	✓	✓	✓
shape		✓		✓
posTag	✓	✓	✓	✓
chunkTag				
entityTag	✓	✓	✓	✓
termType	✓	✓	✓	✓
eventType	✓	✓	✓	
partEvent	✓	✓	✓	

Table 4.2: Features used for the extraction of event triggers.
The feature are shown for the tokens "0" and "-1" for both the training and testing steps. The last three features are the ones to be inferred (cells in gray).

As presented in Table 4.2, the "termType" feature is at the same time known and unknown in the testing step. It is know for the protein terms (which are given), but it is unknown for the remaining entities (events, sites and locations). In this section, we are only concerned on the value for the "termType", which identify the triggers for the events, as well as its type, whether a gene expression or a positive regulation, for instance. The "termType" feature also indicates whether a token is a location or a site. This information was not given and need to be previously extracted.

By considering the features shown in Table 4.2, for the 800 documents in the training set, about 26,788 unique cases were generated. It should be noted that no repetition of cases with the same values for the features are allowed, instead a field for the frequency of the case is incremented to keep track of the number of times that it has appeared during the training phase. The frequency range goes from 1 (more than 22,000 cases) to 238 (one case only).

When a new document is presented to the system in the testing step, it is also read in the forward direction and tokenized. For each token, the system creates a case (the input case) based on the testing features (cf. Table 4.2) and proceeds to search the base for those cases the most similar to the input (cf. Figure 4.12). It should be noted that a token may have more than one input case, depending of the values of the shape feature. The strategy used for the similarity is based on matching the features exactly (cf. 4.2.3.1), i.e., the system tries to find a case with the higher number of features that have exactly the same value of the input case's respective features. The stem is the only mandatory feature whose value must be always matched between the case-problem and the case-solution.

The best case among those retrieved by the system will be the one with the higher frequency. The value of the unknown features will be given by the values of the best case's respective features. If no case is retrieved, because no case matched the mandatory feature ("stem" feature) the token is considered of not being part to any biological event. By repeating this procedure for all the tokens of a document, the latter is then tagged regarding whether it is an event trigger, a location, a site or not participating in any event.

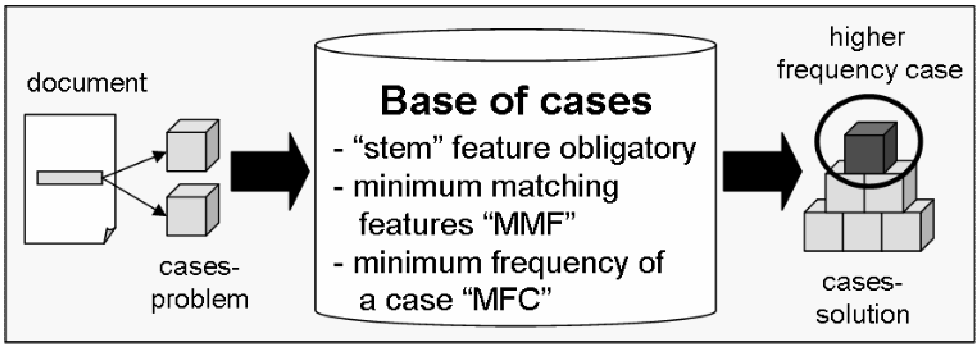


Figure 4.12: Retrieval of a case from the base of cases.
The conditions for selecting a case from the base are shown: the mandatory feature ("stem"), the minimum matching feature and the minimum frequency of the case parameters.

As the BioNLP'09 Event Extraction did not include an evaluation only of the event triggers and the other entities (location and sites), we have carried out our own evaluation of the development dataset in order to check the performance of the

recognition of the named entities, as presented in Table 4.3. Our system is more concerned on obtaining high recall, since an entity not recognized in this step will not be considered further by the extraction of the biological events (cf. 4.4). On the other hand, false positive triggers extracted here may be classified as negative during the next step, if no arguments are found related to it. This is especially true if there is no protein in the same sentence, as the presence of a protein and a trigger are the two mandatory arguments for any type of event.

The overall high recall in Table 4.3 confirms that the extraction of the entities is not a hard task. However, the poor performance of the extraction of some event triggers, such as the binding and the regulatory ones, are in part responsible for the bad performance of the extraction of these events (cf. 4.4). An analysis of the errors is described together with the errors obtained for the relation extraction, in section 4.4.

We have considered two parameters for the retrieval strategy: the minimum matching feature (MMF) and the minimum frequency of the case (MFC). The first one sets the minimum number of non-mandatory features that should be matched between the input case and the cases in the base. It assures that the higher the number of equal features between theses cases, the better is the retrieved case, and more precise is the solution inferred from it.

Events	**(f2m1)**			**(f2m6)**		
	P	**R**	**FM**	**P**	**R**	**FM**
Protein catabolism	70.8	89.5	79.1	69.6	84.2	76.2
Phospholylation	75.0	94.7	83.7	79.1	89.5	84.0
Transcription	22.7	75.9	34.9	36.4	74.6	48.9
Negative regulation	26.4	56.5	36.0	25.3	43.5	32.0
Positive regulation	24.3	63.7	35.2	26.5	59.1	36.6
Regulation	20.8	65.9	31.7	22.1	52.5	31.1
Localization	47.7	79.5	59.6	49.1	66.7	56.5
Gene expression	46.5	83.4	59.7	50.8	80.2	62.2
Binding	29.7	71.1	41.9	29.7	64.4	40.7
Entity	12.5	55.3	20.4	16.8	50.0	25.1
TOTAL	**27.5**	**69.2**	**39.4**	**30.9**	**62.9**	**41.4**

Table 4.3: Evaluation of the event triggers.
Evaluation was performed for the extraction of the trigger event and site/location entities for the development dataset. Results are presented for two situations "f2m1" (MFC=2 and MMF=1) and f2m6 (MFC=2 and MMF=6).

On the other hand, the MFC parameter restricts the cases that are to be considered by the search strategy. It limits them to those with a frequency higher than the value specified by this parameter. The higher the minimum frequency asked for a case, the lower is the number of cases under consideration and the lower is the time for obtaining the case-solution. From the 26,788 cases we have retained during the

training phase, about 22,389 of them appeared just once and would not be considered by the searching procedure if the MFC parameter was set to 2, for example, thus reducing the searching time.

Experiments have been carried out in order to decide the best value for both parameters. A better performance of the system was achieved by setting the MFC to a value higher than 1, i.e., by not considering those cases which appear only once in the training dataset. As expected, experiments have shown that the recall may decrease considerably when restricting the MMF parameter, i.e., few cases may match when requiring the exact matching of many features. Figure 4.13 shows the variation of the F-measure according to both parameters for the values of 1, 3, 4, 5, 6, 7 and 8 for MMF (x-axis); and 1, 2, 5, 10, 15, 20 and 50 for MFC (lines). The experiments were performed with the development dataset.

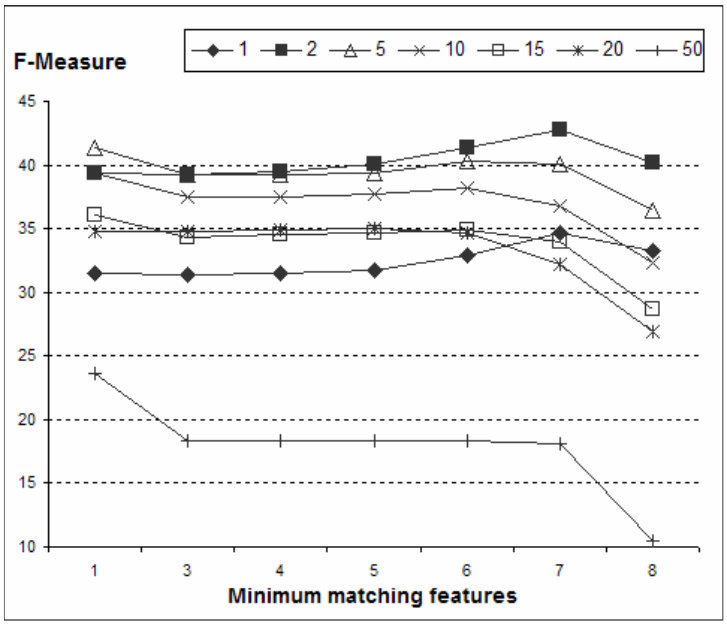


Figure 4.13: Evolution of the F-measure according to the MFC and the MMF parameters.
F-Measure values are plotted for the development dataset in terms of the MFC - minimum frequency of the case (lines) and the MMF - minimum matching feature (x-axis).

Usually, recall is higher for a low value of MFC, as the searching for the best case is carried out over a greater number of cases and the possibility of finding a good case is higher. On the other hand, precision increases when few cases are under consideration by the search strategy, as fewer decisions are taken and the retrieved cases have usually a high frequency, avoiding decision based on "weak" cases of frequency 1, for example.

Figure 4.13 shows that the best value for MFC ranges from 2 to 20 and for MMF from 5 to 7 and the best f-measure result is found for the values of 2 and 6 for these parameters (f2m6), respectively. As these experiments have been carried out after

the deadline of the test dataset, the run that was submitted as the final solution was the one with the values of 2 and 1 for the MFC and MMF parameters (f2m1), respectively. These results are also presented in Table 4.3. Although not being optimal, these were the official results submitted to the challenge.

4.3.2 Recognition of trigger events based on dependency parsing features

In this second approach for the recognition of triggers words, our case is also represented as a window of tokens (cf. 4.2.1.1), but now with the size [-1,1], i.e., it includes both the preceding and the following tokens. This configuration was decided after carrying out some experiments varying the size of the token. The training approach is similar to the first one: it consisted of reading the document only in the forward direction, representing each token as a case according to the features described shown in Table 4.4 and finally saving it in the base of cases. Different to the first approach, when only tokens participating in an event were converted to cases and saved into the database. i.e., those with a non null value for the feature "termType", now all tokens are converted to cases and saved to the base, even the stopwords. The only exceptions are those tokens which correspond to the given proteins, as they are clearly no trigger words, locations or sites.

In this second approach, we also make predictions to the modifier of the event, i.e., whether there is a negation or speculation related to the event. This prediction is performed together with the recognition of the trigger of the event. The features used to represent each element (a token) of the window of tokens are the following:

- lemma of the token (using the Dragon toolkit (Zhou, Zhang et al. 2007));
- part-of-speech tag (provided by the Stanford parser (Klein and Manning 2003));
- distance to the closest protein (number of tokens, in multiples of five);
- direction of the closest protein (whether right or left);
- distance of the dependency tags to the closest protein in multiples of two (de Marneffe, MacCartney et al. 2006);
- type of term ("Entity", "Gene_expression", "Localization", "Binding", "Phosphorylation", "Transcription", "Protein_catabolism", "Regulation", "Positive_regulation" and "Negative_regulation" for the tokens that represent an event, site or location, "noEvent" otherwise) together with the BIEWO tag and the modifier ("none", "Speculation" or "Negation").

The BIEWO tag is used to indicate whether the trigger entity comprises one or more tokens (cf. 3.2). The tags 'B', 'I' and 'E' correspond to the first, inner or last tokens that compose the trigger entity, while the 'W' stands for a trigger composed of one token only and the 'O' for the remaining tokens in the text. Table 4.4

summarizes the features that have been considered for each token of the window and whether it is mandatory or not.

Features	Type	Token -1	Token 0	Token +1
Lemma	Nominal	×	✓	×
Part-of-speech tag	Nominal		✓	×
Direction closest protein	Nominal		✓	
Distance closest protein	Numerical		×	
Distance dependency tag closest protein	Numerical		×	
Type of term + BIEWO tag	Nominal	×	?	
Modifier	Nominal	×	?	

Table 4.4: Features of the biological event triggers.
Features marked with an "×" are the ones considered for the corresponding token, those with a "✓" are the mandatory ones and the unknown features are identified with a "?".

As the type of the term and the modifier are unknown features, their values for the "-1" token are obtained from the cases retrieved in a previous step chosen for this token. If no case is retrieved for an input case, we assign the values "noEvent" for the type of term feature and "none" for the modifier feature, i.e., the corresponding token is not recognized as an event trigger. This situation can only happen if no case matches the mandatory features.

Tokens assigned a value of "Entity" for the type of term feature are recognized as a site or location, otherwise the token is recognized as the corresponding trigger event, e.g. gene expression, transcription. The searching strategy used for the extraction of the event triggers, sites and locations was the exactly matching features (cf. 4.2.3.1).

Events	Recall	Precision	F-Measure
Gene expression	82.1	64.1	72.0
Transcription	60.6	41.6	49.3
Protein catabolism	84.2	72.7	78.0
Phosphorylation	92.1	74.5	82.4
Localization	66.7	68.4	67.5
Binding	57.8	57.5	57.6
Regulation	44.4	33.9	38.4
Positive regulation	49.4	41.5	45.13
Negative regulation	47.3	38.9	42.7
Entity	44.7	36.5	40.2
Total	**58.2**	**48.3**	**52.7**

Table 4.5: Results for the named-entity recognition task for the development dataset.
The "Entity" category includes all the other entities distinct from events, such as sites and locations.

Again, we have carried out our own evaluation of the development dataset in order to check the performance of the recognition of the named entities (triggers, sites and locations). Results are shown in Table 4.5. This classifier is more concerned with obtaining high recall since an entity not recognized in this step will be not considered by the relationship extraction classifier (cf. 4.4.2). On the other hand, false positive triggers extracted here may be classified as negative during argument detection, especially if there is no protein in the same sentence or even due to the case extracted as solution for the corresponding context not being an event.

The overall high recall in Table 4.5 confirms that the extraction of the entities is not as hard a task as relationship extraction. An analysis of the errors obtained with the improved methodology is descried together with the errors obtained for the relationship extraction in section 4.4.2).

Table 4.6 shows a comparison of the results which were submitted to the challenge (cf. 4.3.1), which were only based on shallow linguistic methods, and those obtained with the improved version. The precision of the system has significantly improved without much loss of the recall.

	Recall	**Precision**	**F-Measure**
Challenge (f2m1)	69.2	27.5	39.4
Challenge (f2m6)	62.9	30.9	41.4
Improved methodology	58.2	48.3	52.7

Table 4.6: Evolution of the results for the extraction of the event triggers.
Comparison of the results obtained during the challenge with those of the improved methodology.

4.4 Extraction of Biological Events

In this section we will describe the two methodologies we propose for the extraction of the events, i.e., the relationships between the event trigger (previously extracted in sections 4.3.1 and 4.3.2) and its many arguments. The arguments may be a protein (which was given), a site or location, also previously extracted in sections 4.3.1 and 4.3.2.

Here we propose two methodologies, one based on manual rules, which participated in the BioNLP'09 Event extraction challenge (Neves, Carazo et al. 2009) (cf. 4.4.1), and a second one (cf. 4.4.2) which uses the general methodology described in section 4.2 for case-based reasoning. Here we will only describe those details which are specific of event extraction task.

4.4.1 Extracting biological events based on manual rules

The aim of this task is to associate the trigger words with the arguments that play a role in a biological event. For this first approach, the event triggers are the ones extracted using the shallow linguistic methodology (cf. 4.3.1).

Our approach proposes to extract the arguments incrementally, using the trigger words are the starting point. The order by which the arguments are extracted from the text is the following: theme, theme2, cause, site and location. The rules are based on the values assigned for the three unknown events in section 4.3.1: "termType", "eventType" and "partEvent". By analyzing some of our false negatives returned in the development dataset, we have learned that few events are associated to arguments present in a different sentence and although we are aware of some few cases, we have decided to restrict the searching to the sentence boundaries in order to avoid a high number of false positives. Figure 4.14 resumes the rules for each of the arguments.

Themes: The candidates for the "theme" argument are the annotated proteins as well as the events themselves, in the case of the regulation, positive regulation and negative regulation events. The first step is then to try to map each event to its theme and in case that no theme is found, the event trigger is considered as wrong, i.e., a false positive from the trigger extraction step.

The searching strategy starts from the event trigger and it consists of reading the text in both directions alternatively, one token in the forward direction followed by one token in the backward direction until a candidate for them is found (cf. Figure 4.14). The system halts if the end of the sentence is found or if the specified number of tokens in each direction is reached, which is 20 for the theme.

Theme2: regarding the second theme, which occurs only in the binding events, a similar searching strategy is carried out, except that now the system reads up to 10 tokens in each direction, starting from the theme which was previously extracted.

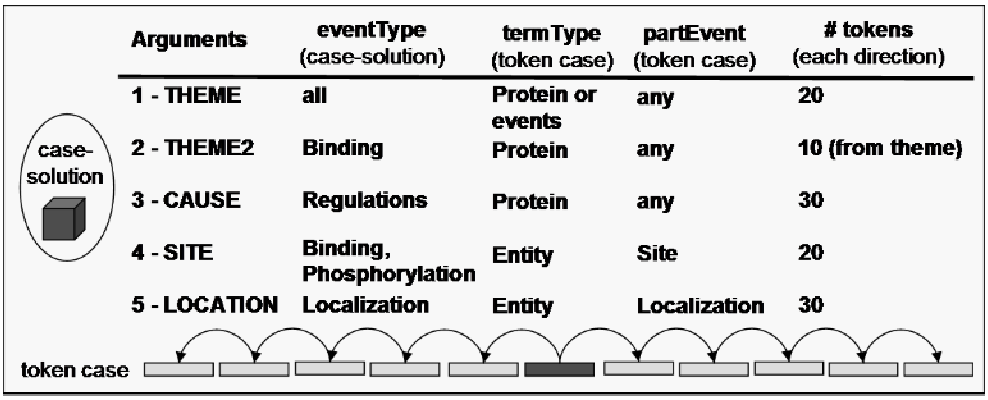


Figure 4.14: Summary of the rules for the extraction of each type of argument.
Summary of the rules used for the extraction of the theme, theme2, cause, site and location. The first column shows the name of the arguments and the following columns present which values to expect for the features "eventType", "termType" and "partEvent". The last column show the maximum number of tokens considered while searching for the event's arguments.

Cause: The candidates for the "cause" arguments are also the annotated proteins and another event, in the case of the regulatory events. Starting from the event trigger, a similar searching strategy is carried out here, this time restricted up to 30 tokens in each direction and to the boundaries of the same sentence. This procedure is carried out only for the regulation, positive regulation and negative regulation events. We have defined an extra restriction that the candidates should not be the protein already assigned as theme for the event. If no candidate is found, the system considers that there is no cause associated to the event.

Site and Location: Here the candidates are the tokens tagged with the values of "Entity" for the termType feature, and "Site" and "Location" for the "partEvent" feature, respectively (cf. 4.3.1). The searching for the site is carried out only for the binding and phosphorylation events and the searching for the location only for the localization event. This procedure is also restricted to the sentence boundaries and up to 20 and 30 tokens, respectively for the site and location, starting from the event trigger. Once again, if not candidate is found, the system consider that there is no site or location associated to the event.

Some experiments have been carried out with the development and the blind test datasets as well as an analysis of the false negatives and false positives. Results here will be presented in terms of precision, recall and f-measure for tasks 1 and 2 proposed in the BioNLP Shared task (cf. B.4), i.e., for the extraction of simple events (only one arguments) and for the complex ones (many arguments). Table

4.7 and Table 4.8 resume the results obtained for the test dataset. We show results for two configurations for the extraction of event triggers (cf. 4.3.1): the one that was submitted (f2m1), and the best one (f2m6) after carrying out the experiments described in section 4.3.1.

Tasks / Results		Recall	Precision	F-measure
Task 1	**(f2m1)**	28.63	20.88	24.15
	(f2m6)	27.18	23.92	25.45
Task 2	**(f2m1)**	25.02	18.32	21.15
	(f2m6)	24.49	21.63	22.97

Table 4.7: Results for the test dataset.
Results are shown for the tasks 1 and 2 of the BioNLP Shared Task using the "f2m1" and "f3m6" configurations for the event trigger extraction.

An automatic analysis of the false positives and false negatives has been performed for the development dataset and for the configuration (f2m1), the one which was submitted for the BioNLP Shared Task challenge. The errors consist on a total of 2502 false positives and 1300 false negatives. We have found out that the mistakes are related mainly to the retrieving of the case from the database and to the mapping of an event to its arguments.

Results / Events	(f2m1)			(f2m6)		
	p	**r**	**fm**	**p**	**r**	**fm**
Protein catabolism	78.6	55.0	64.7	71.4	55.6	65.5
Phosphorylation	49.6	56.1	52.7	46.0	55.2	50.2
Transcription.	48.9	19.8	28.1	38.7	29.6	33.5
Negative regulation	9.8	7.9	8.8	7.9	7.7	7.8
Positive regulation	10.0	6.6	7.9	10.2	8.0	9.0
Regulation	8.6	4.5	5.9	7.5	5.3	6.3
Localization	28.2	42.9	34.0	23.3	48.9	33.3
Gene Expression	51.8	55.1	53.4	52.6	61.2	56.6
Binding	19.5	12.1	14.9	22.4	14.4	17.5

Table 4.8: Results for each event for the test dataset.
Detailed results for each type of event are shown for task 2 of the BioNLP Shared Task for the f2m1 and f3m6 configurations for the event trigger extraction.

The analysis of errors includes those obtained for the task of recognizing the event triggers (cf. 4.3.1). Figure 4.15 and Figure 4.16 show the percent contribution of each class for the false positives and false negatives, respectively. The automatic analysis of the false positive and false negative mistakes is a hard task since no hint is given for the reason of the mistake by the evaluation system, whether due to the event type or to wrong theme, an incorrectly association to an event or even a missing cause or site. The mistakes have been classified in seven groups which are described below.

Events composed of more than one token (1): this mistake happens when the system is able to find the event with its correct type and arguments but with only part of its tokens, such as "regulation" instead of "up-regulation" and "reduced" or "levels" instead of "reduced levels", both found in document 10411003. This is mainly due to our tokenization strategy of separating the tokens according to all punctuation and symbols (including hyphens) and also due to the evaluation method that seems not to consider alternatives to the text of an event. This mistake always results in one false positive (e.g., "regulation" or "reduced") and one false negative (e.g., "up-regulation" or "reduced leves").

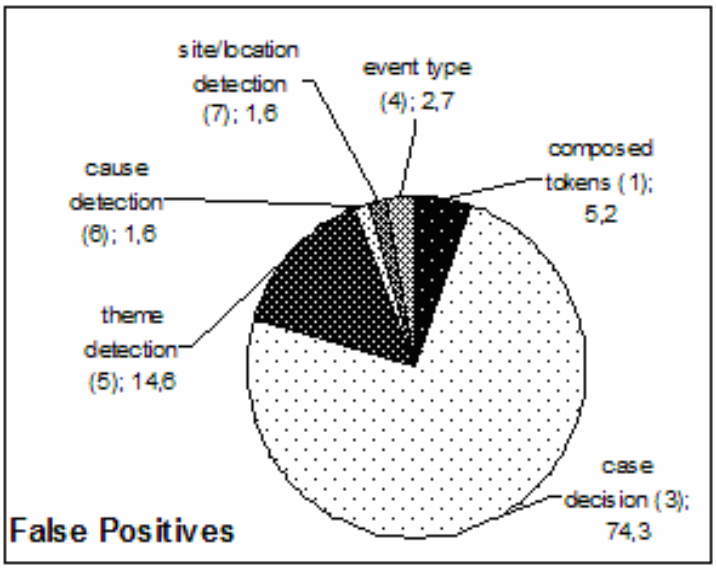


Figure 4.15: False positives for the event extraction using manual rules.
The contribution in percent of each error in the false positives is shown. For each mistake, a short name and the number of the type of error is included.

Events and arguments in different sentences of the text (2): as we have previously discussed, our arguments searching strategy is restricted to the boundaries of the sentence. Some examples of mistakes dues to this restriction can be found in document 10395645 in which two events of the token "activation [1354-1364]" is mapped to the themes "caspase-6 [1190-1199]" and "CPP32 [1165-1170]", both located in a different sentence. This error usually affects only the false negatives but may cause also a false positive if the system happens to find a valid (and wrong) argument in the same sentences for the event under consideration.

Decision for a case (3): this error is due to the retrieval of a wrong case from the base. Case-based reasoning is only used for the extraction of the event trigger, so this mistake can happen in two situations: when the system fails to find any case for a token which represents an event (false negative) or when a case (representative of a event trigger) is retrieved for a token which is no event trigger

at all (false positive). The first situation is only dependent of the searching strategy and its two parameters (MMF and MFC) (cf. 4.3.1), while the second one is also related to the post-processing step, if the latter succeeds to find a theme for the incorrectly extracted event trigger. An example of a false negative that falls in this group is "dysregulation [727-740]" from document 10229231 that failed to be recognized as an event trigger. Regarding the false positives, this class of mistake is the majority of them and it is due to the low precision of the system that often is able to retrieve cases associated to tokens which are not events at all, such as the token "transcript [392-402]" of document 10229231. It should be noted that the incorrect recognition of a token as an event trigger does not result in a false positive a priori, but only if the post-processing step happens to find a valid theme to it, a mistake further described in group 5.

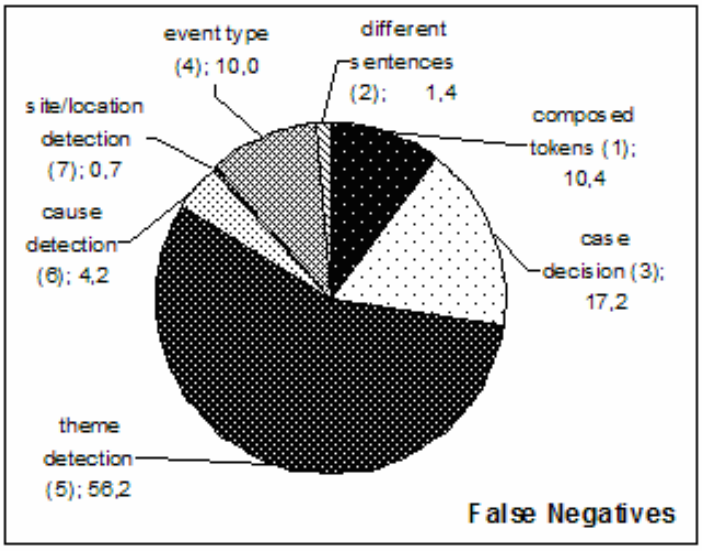


Figure 4.16: False negatives for the event extraction using manual rules.
The contribution in percent of each error in the false positives is shown. For each mistake, a short name and the number of the type of error is included.

Wrong type of the event (4): this class of mistake is also due to the retrieval of the wrong case. The difference here is that the token is really an event, but the retrieved case belongs to the wrong type, i.e. it has a wrong value for the "eventType" feature (cf. 4.3.1). The causes of this mistake are many, such as, the selection of features or the value of the MFC parameter that may lead to the selection of a wrong but more frequent case. We also include in this group the few false negatives mistakes in which a token is associated to more than one type of event in the gold-standard, such as the token "Overexpression [475-489]" from document 10229231 which consists of both a Gene Expression and a Positive Regulation event. However, our methods assign just one case (one type of event) per token. One way to overcome this problem would be to allow the system to associate more than one case to a token, taking the risk of decreasing its precision.

Theme detection (5): in this group falls more than half of the false negatives and we include here only those mistakes related to the theme argument for event whose trigger has been correctly extracted. These mistakes may be due to a variety of situations related to the theme detection, such as: the association of the event to another event when it should have been done to a protein or vice-versa (for the regulation events); the mapping of a binding event to only one theme when it should have been to two themes or vice-versa; the association of the event to the wrong protein theme, especially when there is more than one protein nearby; and even not being able to find any theme at all. Also, half of theses mistakes happen when an event is associated to more than one theme separately, not as a second theme. For example, the token "associated [278-288]", from document 10196286, is associated in the gold standard to three themes – "tumor necrosis factor receptor-associated factor (TRAF) 1 [294-351]", "2 [353-354]" and "3 [359-360]" – and we were only able to extract the first of them.

Cause detection (6): similar to group 5, these mistakes happens when associating a cause to an event (regulation events only) when there is no cause related to it or vice-versa. For example, in document 10092805, the system has correctly mapped the token "decreases [1230-1239]" to the theme "4E-BP1 [1240-1246]" but also associated to it an inexistent cause "4E-BP2 [1315-1321]". The evaluation of Task 2 does not allow the partial evaluation of an event and therefore a false positive and a false negative would be counted for this example.

Site/Location detection (7): this error is similar to the previous one but related only to binding, phosphorylation and localization events. Here the system fails to associate a site or a location to an event or vice-versa. For example, in document 10395671, the token "phosphorylation [1091-1106]" was correctly mapped to the theme "Janus kinase 3 [1076-1090]" but was also associated to an inexistent site "DNA [1200-1203]". Once again, the evaluation of Task 2 does not allow the partial evaluation of the event and a false positive and a false negative would be returned.

Regarding our results, they show that our system has performed relatively well using a simple methodology of a CBR-based extraction of the event triggers (cf. 4.3.1) together with some manual rules for the association of its arguments. The analysis of the mistakes presented here confirms the complexity of the tasks proposed during that challenge, but it also show that the extraction of the event triggers (cf. Table 4.3) is an easier task.

We believe that the part of our system which requires most our attention is the retrieval of the case-solution and the theme detection in the post-processing step. Improvements in this line could increase the precision and recall, respectively. The decision of searching for a second theme and associating a single event separately

to more than one theme is hard to be accomplished by manual rules and could better be learned automatically using a machine learning algorithm. Such a methodology is proposed below and it was developed after the challenge.

4.4.2 Extracting biological events using case-based reasoning

In this section we describe an improved methodology for the extraction of biomedical events. We utilize a cased-based reasoning classifier different from the one used for named-entity recognition (cf. 4.3.2). The text of a document is read in the forward direction during both training and testing steps. A case here is represented as a context (cf. 4.2.1.2) whose length is automatically defined by some predefined tokens of the sentence, i.e., the entities that might be involved in the event.

Since the event extraction is a complex task, many changes were required in the general algorithm (cf. 4.2) in order to take the particularities of the domain into account. The extraction of the trigger events and those entities which are not given (triggers, sites and locations) is carried out using the methodology described in section 4.3.2. The tokens identified as "Entity" by this classifier are used as arguments for the events, along with the given annotated proteins. The construction of the bags of entities (cf. 4.2.2) is rather simple, given the type of entity of the token (cf. Figure 4.9 in page 86): proteins go to the "Proteins" bag, sites and locations to the "Entities" bag and event triggers to the "Events" bag.

Automatic generation of the contexts is one of the most complex tasks here, because the corpus includes many types of events (regulation, localization, gene expression, binding, etc.) that may be composed of distinct numbers of arguments from distinct types. For example, a binding event may have one, two or three themes, and any of them may or may not contain an associated site. Fortunately, the problem is limited to the number of entities inside each bag. For example, if there are three proteins (p1, p2, p3) in the "bag of proteins" and one site in the "bag of sites", the following contexts could be generated for a binding event, grouped according to the number of arguments in the context:

- trigger-theme (2 arguments): trigger-p1, trigger-p2, trigger-p3 (total of 3 events);
- trigger-theme-site (3 arguments): trigger-p1-s, trigger-p2-s, trigger-p3-s (total of 3 events);
- trigger-theme-theme2 (3 arguments): trigger-p1-p2, trigger-p2-p1, trigger-p1-p3, trigger-p3-p1, trigger-p2-p3, trigger-p3-p2 (total of 6 events);
- and so on.

Therefore, the number of candidate events for a single event trigger may quickly increase depending on the number of proteins, events and sites/locations in the

bags of entities. Additionally, when evaluating the shared task, the order of themes in the event matters. For example, the event "trigger-p1-p2" is different from the event "trigger-p2-p1", and both possibilities should be taken into account by the system.

Number of arguments	EXP, TRA, CAT	PHO	LOC	BIN	REG, POS, NEG
1	TH	TH	TH	TH	TH
2	-	TH-ST	TH-TL TH-AL	TH-TH2 TH-ST	TH-CA TH-ST TH-CS
3	-	-	-	TH-TH2-TH3 TH-ST-TH2 TH-TH2-ST2	TH-CA-ST TH-CA-CS
4	-	-	-	TH-ST-TH2-ST2 TH-ST-TH2-TH3 TH-TH2-ST2-TH3 TH-TH2-TH3-ST3	-
5	-	-	-	TH-ST-TH2-ST2-TH3 TH-ST-TH2-TH3-ST3 TH-TH2-ST2-TH3-ST3	-
6	-	-	-	TH-ST-TH2-ST2-TH3-ST3	-

Table 4.9: Combinations of arguments for each type of event.
For each type of events, the combinations of arguments that may appear and which have been taken into account are shown. The combinations are also classified according to the number of arguments in its composition (1 to 6). The following abbreviations are used for the arguments: TH=theme, TH2=theme2, TH3=theme3, ST=site, ST2=site2, ST3=site3, TL=toLoc, AL=atLoc, CA=cause, CS=csite. The trigger is not shown above, although it is present for all events.

In summary, the automatic generation of contexts depends on the type of the entity to be generated but it is limited to the entities contained in the bags of arguments. An overview of the combination of arguments to be considered according to the type of event is presented in Table 4.9. The complexity of some features, especially of the binding feature, can be clearly appreciated in it.

Because of time constraints and in order not to generate extremely long contexts which would not be very helpful during the testing step, we have limited the size of the context to 20 tokens. No experiments have been carried out taking the surrounding right and left tokens into account. The contexts are always delimited by the right-most and left-most entities in its composition. A practical example of the automatic generation of the context is shown in Figure 4.17. In this figure, for

the sentence "Interleukin-10 inhibits expression of both interferon alpha- and interferon gamma- induced genes by suppressing tyrosine phosphorylation of STAT1", the third context defined for the "phosphorylation" trigger is composed by the protein "STAT1" and the site "tyrosine", is the representation of the text "tyrosine phosphorylation of STAT1". The rest of the sentence is not taken into account for this particular event.

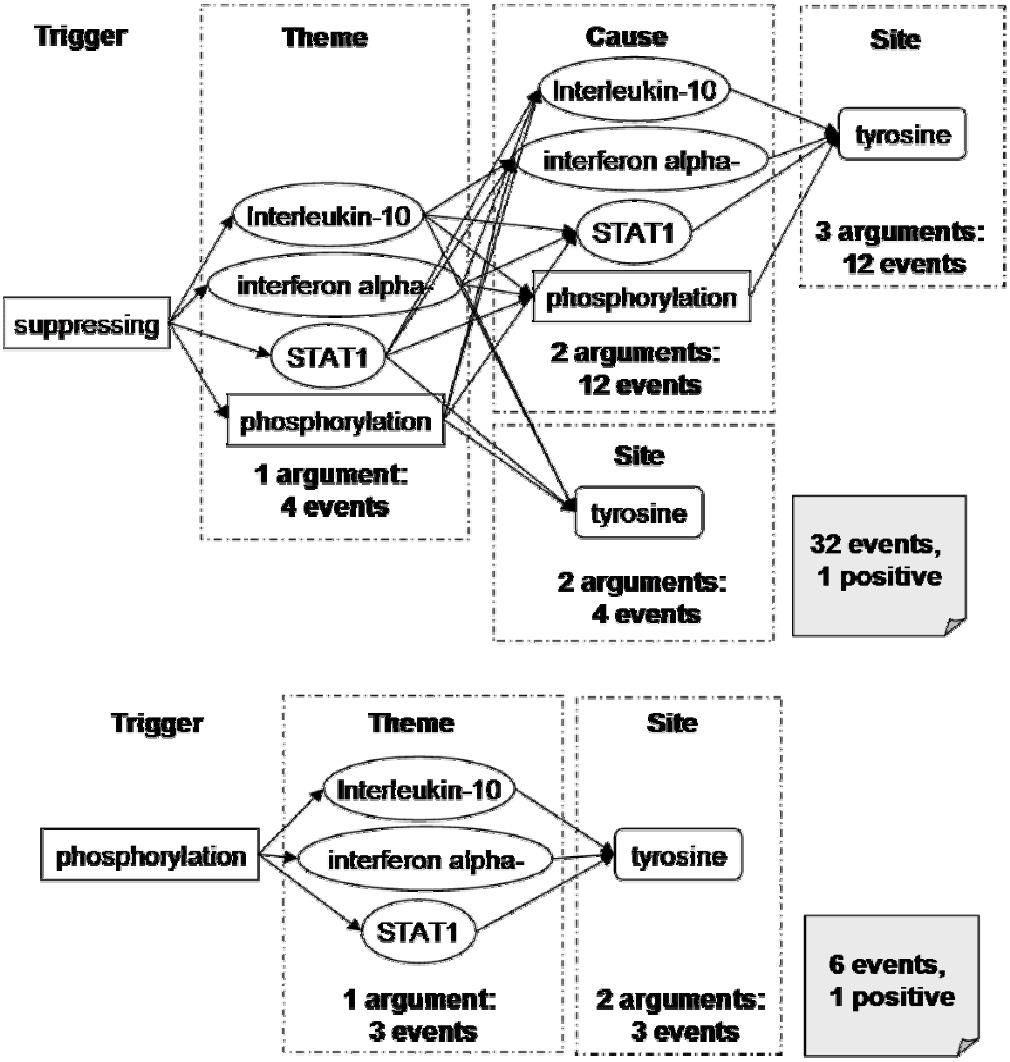


Figure 4.17: Example of the automatically generated contexts for events.
Contexts are automatically generated for the triggers "suppressing" (a negative regulation event) and "phosphorylation" (a phosphorylation event) using the bags-of-arguments shown in Figure 4.9. Proteins are identified as ellipses, triggers as rectangles and other entities (sites) as rounded rectangles. The positive events are those present in the training dataset. All combinations of the arguments are generated, according to the rules specified by the type of event. For instance, the negative regulation event may take proteins and other events for the "theme" or "cause" arguments.

The cases that represent an event context are composed of the features listed in Table 4.10. The category feature indicates whether the relationship is true or false, given for the training dataset and it unknown for the cases in the development and test datasets. The second feature at context level is the one representing the distinct types of entities that compose the context. The entities involved in the context may be of various types ("Protein", "Entity", another event, etc) depending on the type of event. This feature also takes into account the order in which they appear in the text and it comprises not only the type of entity but also the role it plays in the context. This is because a determined type of entity, for example a gene expression event, may be the trigger role in a certain event and the theme or the causal role in a regulatory event. Some examples of the values of this feature may take are the following:

- "AtLoc:Entity,Trigger:Localization,Theme:Protein",
- "Site:Entity,Theme:Protein,Trigger:Binding",
- "Site:Entity,Cause:Protein,Trigger:Positive_regulation,Theme:Regulation"

Elements	Features	Type	Status
Context	Category	Nominal	**?**
	Entities of the context	Nominal	✓
Context Items	Part-of-speech tag	Nominal	×
	Type of entity	Nominal	×
	Role	Nominal	×

Table 4.10: Features of the event extraction classifier.
Features marked with an "x" are the ones considered for the corresponding element, the one marked with a "check" is the mandatory and the feature-problem is identified by a question mark.

Being the type of entity a mandatory feature, the searching for cases is restricted to the ones with the same value for this feature. The features for the context items include the part-of-speech tag of the corresponding token, which was extracted using the Stanford parser (cf. C.4), the type of entity (e.g., "Protein", "Entity", "Localization") and the role of the item in the context (e.g., "Trigger", "Cause").

In conclusion, contexts classified as positive are selected as the event candidates. However, more than one event candidate may be extracted for each event trigger, so some post-processing procedures are needed for choosing the final events present in a given sentence. These post-processing procedures are also due to the restrictions in the format of the output file of the BioNLP'09 Event Extraction organization. These are the post-processing procedures:

Joining consecutive trigger tokens: as the contexts are generated on the basis of single token triggers, this step tries to join trigger tokens which appear

consecutively in the text and whose events are of the same type. More than two tokens may be joined together, as long as these conditions are satisfied.

Cleaning cross reference events: the system filters regulatory events that have cross references, i.e., we check if a certain event E1 has a reference to an event E2 in its theme or cause, and at the same time, if event E2 has a reference to E1 in its theme or cause. We clean the cross-reference by removing the whole event or cleaning the cause argument to the event that has received more votes.

Filtering equivalent events: Although we are allowed to provide more than one event associated with the same trigger, those must be of different types, if referred to the same theme, or must refer to different themes if they are of the same type. As we make no restriction in generating of the contexts, more than one event of the same type and referring to the same theme may have been classified as positive. However, there is always a difference in any of the optional arguments, such as different causes, sites, extra themes, etc. In this step, for events with the same type, trigger and theme, only one is selected, the one that has received most votes or has most extra arguments (besides the theme).

	Development			Test		
	Recall	**Precision**	**FM**	**Recall**	**Precision**	**FM**
Total Simple Events	72.99	54.12	62.15	57.19	50.71	53.76
Binding	19.28	37.21	25.40	22.77	44.13	30.04
Total events	56.44	51.65	53.93	49.38	49.93	49.65
Total Regulation Events	20.36	22.95	21.58	12.04	17.48	14.26
Total Modifiers	7.43	8.57	7.96	5.52	10.17	7.15
Total	**33.65**	**35.21**	**34.41**	**27.04**	**33.91**	**30.09**

Table 4.11: Results of Task3 for the Event Extraction corpus.
The results were evaluated using the approximate span matching/approximate recursive matching method. "Total Simple Events" include the following: gene expression, transcription, protein catabolism, phosphorylation and localization. "Total events" include all simple events and the binding event. "Total Regulation Events" refers to regulation, positive regulation and negative regulation events. "Total Modifiers" refers to evaluation of the speculation and negation modifiers. Finally, the last line of the table is the evaluation of all the above together.

The evaluation of the methodology was carried out with the development and blind test datasets (cf. B.4). Here we present the results for the "approximate span matching/approximate recursive matching" method, which was chosen by the shared task organization for comparison among the participants' results (Kim, Ohta et al. 2009). Table 4.11 shows an overview of the precision, recall and f-measure for Tasks 1, 2 and 3 (cf. B.4), for both the development and blind test datasets. These results show that there is no huge difference between the results of the two datasets and prove that the algorithm is not tuned for the development dataset.

Especially for simple events, the recall of the system is much higher than the corresponding precision.

The results show that the more complex the type of event is, the worse is their performance, since more arguments need to be matched. This is the case for the Binding event, one of the most complex events under consideration here, as it allows up to three themes in its arguments and their respective sites. Most of the time, it is hard for the system to decide whether the nearby proteins are related to the trigger event and also whether they should be treated as two distinct themes of a single event or main themes of two separate events. The performance of this event is also highly dependent on the "Entity" recall (cf. Table 4.5 in page 96). Finally, regulation events may be the most complex ones. The main problem is that the theme and cause arguments can be mapped to an event or a protein and if the mapped event is not correctly extracted, the theme or cause argument mapped to it is also considered incorrect during the evaluation.

Task 1				
Groups	**Simple Ev.**	**Binding**	**Regulation**	**All**
CNBMadrid	48.35 (16/24)	25.36 (12/24)	9.67 (13/24)	24.15 (18/24)
CBR	55.01 (10/25)	30.44 (7/25)	20.83 (7/25)	36.13 (7/25)
UTurku	70.21 (1/24)	44.41 (1/24)	40.11 (1/24)	51.95 (1/24)
Task 2				
Groups	**-**	**-**	**-**	**All**
CNBMadrid	-	-	-	21.15 (6/6)
CBR	-	-	-	34.34 (2/7)
UT+DBCLS	-	-	-	43.12 (1/6)
Task 3				
Groups	**Negative**	**Speculative**	**-**	**-**
CBR	10.66 (3/7)	2.86 (7/7)	-	-
ConcordU	23.13 (1/6)	25.27 (1/6)	**-**	**-**

Table 4.12: Comparative results for the Event Extraction corpus.
A comparison of the results for Tasks 1, 2 and 3 is shown in regard to our previous participation in the BioNLP Shared Task challenge and to the best results that have been published in the latter.

We compare our results to the best ones for each of the tasks. The results are the ones that were published during the BioNLP Shared Task workshop (Kim, Ohta et al. 2009), as presented in Table 4.12. Our participation in the challenge is identified as the "CNBMadrid" group, also presented in the table under this name. We use "CBR" to label the results obtained with the methodology proposed in this section, while the results from other groups that we compare are identified by their names in the challenge. The position of the results, based on the f-measure value and in relation to the total number of results, is presented beside the F-measure values. In the case of the "CBR" results, we suppose that the list is composed by

one more participant, as these results were not originally in the challenge list of results. More detailed results for the development and test datasets, for each type of event, are presented in Table 4.13 below for both the development and test datasets.

Type of event	Development			Test		
	Recall	**Precision**	**F-M**	**Recall**	**Precision**	**F-M**
EXP	79.69	59.52	67.02	62.47	59.81	61.11
TRA	64.63	41.41	50.48	34.31	29.38	31.65
CAT	85.71	54.55	66.67	71.43	34.48	46.51
PHO	70.21	50.00	58.41	65.93	49.44	56.51
LOC	58.49	45.59	51.24	43.10	54.35	48.08
Sub-Total 1	**72.99**	**54.12**	**62.15**	**56.85**	**53.29**	**55.01**
BIN	19.28	37.21	25.40	22.77	45.93	30.44
Sub-Total 2	**56.44**	**51.65**	**53.93**	**49.12**	**52.41**	**50.71**
REG	15.61	17.42	16.46	12.71	17.37	14.68
POS	22.01	25.66	23.69	19.94	27.67	23.18
NEG	19.39	19.90	19.64	16.36	24.12	19.50
Total Regulat	**20.36**	**22.95**	**21.58**	**17.85**	**25.02**	**20.83**
Negation	7.48	9.30	8.29	9.25	12.58	10.66
Speculation	7.37	7.41	7.39	1.92	5.56	2.86
Modifiers Total	**7.43**	**8.57**	**7.96**	**5.75**	**10.39**	**7.40**
Total	**33.65**	**35.21**	**34.41**	**29.61**	**37.67**	**33.16**

Table 4.13: Results for Task 3, development dataset
Details according to the event are presented for the development and test dataset. Abbreviation used for the events are the following: "EXP" for gene expression, "TRA" for transcription, "CAT" from protein catabolism, "PHO" for phsophorylation,"LOC" for localization, "BIN" for binding, "REG" for regulation, "POS" for positive regulation and "NEG" for negative regulation.

Table 4.14 show details of the performance of the methodology for some pairs of event trigger and arguments, in order to check those which are more easily and more hardly extracted. For example, the extraction of the "theme" argument is harder for the regulatory events than for the others, as it may be represented as a protein or another event. The same happens for the "cause" argument, which also has a poor performance. The harder argument is the "site" and "csite", as they are not given and are dependent on the performance of the named-entity recognition step (cf. 4.3.2).

An automatic analysis of the errors for the development dataset resulted in a total of 1119 false positives and 1144 false negatives. We found that about half the false positives and more than 60% of the false negatives are related to the extraction of the trigger token (cf. 4.3.2). This may happen because the classifier sometimes fails to retrieve a case for the token or because cases were found but the voting scheme decided that the token is not an event. The ambiguity among the event trigger tokens is huge as some words that one would related only to a specified

type of event, such as "expression" to the gene expression event, happen to be associated with transcription, localization or positive regulation events or even to no event at all.

Event-Arg Type	Recall	Precision	F-measure
Gene expression – Theme	62.47	59.81	61.11
Transcription - Theme	34.31	29.38	31.65
Protein catabolism – Theme	71.43	34.48	46.51
Binding – Theme	31.91	60.48	41.78
Binding – Site	0.00	0.00	0.00
Phosphorylation – Theme	64.44	48.33	55.24
Phosphorylation – Site	37.50	63.64	47.19
Localization – Theme	43.10	54.35	48.08
Localization – AtLoc	13.51	41.67	20.41
Localization – ToLoc	28.57	57.14	38.10
Regulation – Theme	18.28	23.94	20.73
Regulation – Cause	5.08	20.00	8.11
Regulation – Site	20.00	25.00	22.22
Regulation – CSite	0.00	0.00	0.00
Positive regulation – Theme	26.18	35.24	30.04
Positive regulation – Cause	14.04	45.83	21.50
Positive regulation – Site	16.13	35.71	22.22
Positive regulation – CSite	0.00	0.00	0.00
Negative regulation – Theme	19.57	27.48	22.86
Negative regulation – Cause	0.00	0.00	0.00
Negative regulation – Site	22.22	33.33	26.67
Negative regulation – CSite	0.00	0.00	0.00
All Total	**33.58**	**43.89**	**38.05**

Table 4.14: Detailed results by argument for the Task 2.
Results for the combination of some event type and arguments are shown.

Mistakes in which triggers comprises more than one token are also related to the trigger extraction. We use the BIEWO tag (cf. 3.2.1) in order to take these cases into account, but sometimes the system could only find part of the trigger token, such as "appeared" instead of "appeared normal" (document 10092801). This mistake corresponds to about 4% and 2% of the false positives and false negatives, respectively. Additionally, the named-entity recognition classifier sometimes assigns the wrong type of event to the token, owing to the wrong selection of the case from the base. We also include here a few false negative mistakes attributable to tokens that are associated with more than one event of different types in the gold-standard, such as the token "Overexpression" in document 10229231, which is mapped to two events, a gene expression and a positive regulation. Nonetheless, our algorithm allows more than one type of event for a token to be selected through

the voting scheme. This mistake corresponds to about 7% of the false positives and false negatives.

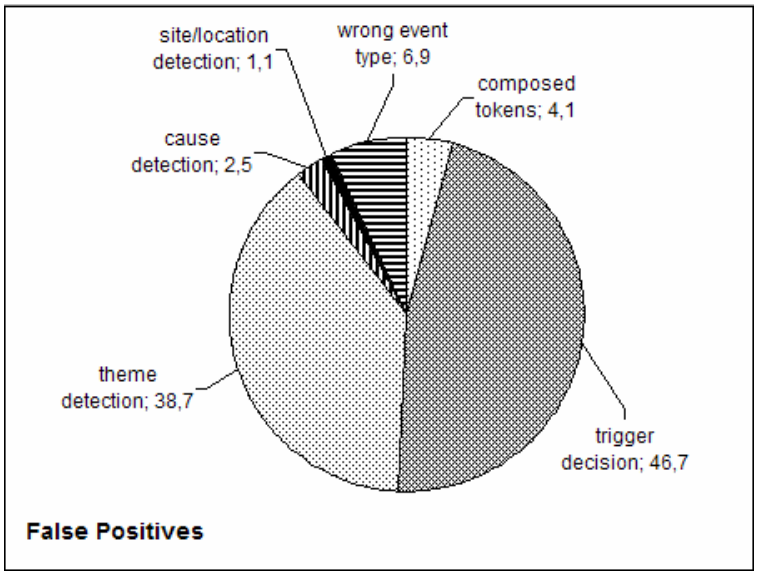


Figure 4.18: Distribution of the error for the false positives.
Contribution of each type of error for the false positives mistakes is shown.

We classified the remaining mistakes according to problems related to the mapping of the remaining arguments (theme, cause, site and location). Being a mandatory argument, theme detection corresponds to almost 40% of the false positives and 23% of the false negatives. The mistakes related to theme detection may be due to a variety of situations such as: the association of a regulatory event with another event when it should have associated with a protein (or vice versa); the mapping of a binding event to one theme only when it should have been mapped to two themes (or vice versa); the association of the event with the wrong protein theme or the wrong event theme, especially when there is more than one nearby; and even being unable to find any theme at all.

The errors related to the detection of the "cause· argument constitute only 2% of the false positives and false negatives, and they are due to associating a cause with an event or protein (for regulatory events) when there is no cause at all to it, or vice versa. Finally, errors in site and location detection correspond to just 1% of the false positives and false negatives, and they happen when the system fails to associate a site or a location with an event, or when it is done when it should not have been done. Figure 4.18 and Figure 4.19 illustrate the distribution of each type of error for the false positives and false negatives.

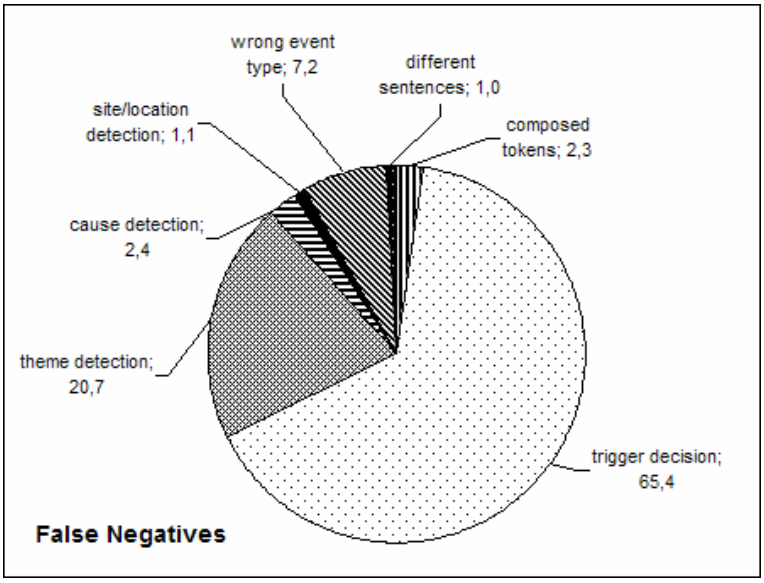


Figure 4.19: Distribution of the error for the false negatives.
Contribution of each type of error for the false negative mistakes is shown.

The errors showed above are detailed in Table 4.15 and Table 4.16 (in page 115 and page 116, respectivelly) according to the type of event. A description of each of the errors of the tables above is showed below. We separate the errors related to the named-entity recognition classifier to the ones related to the extraction of the arguments. Some errors may be represented by the same area in Figure 4.18 and Figure 4.19 because they are of the same type, therefore, the reference to the error in those figures are also included in these tables (between parenthesis). Some errors may be related just to the false positives or to false negatives.

These are the errors related to the extraction of the event triggers:

- composed_token_event (composed tokens): the system was able to recognize only part of the trigger token when it is composed of more than one token.
- wrong_type_event (wrong event type): the system has assigned the wrong type of event to the trigger token.
- no_case_found (trigger decision): the system was not able to find a valid case for the trigger token and it has been classified as not being an event.
- trigger_no_event (trigger decision): the system has been able to retrieve a case for the trigger token but the voting scheme has decided as it not being an event.
- token_no_event (trigger decision): the token has been classified as an event when it is not.

And these are the errors related to the extraction of the arguments:

- argument_no_event (trigger decision): the trigger token was recognized as an event but the voting scheme of the relation extraction classfier has classified all the corresponding contexts related to it as negative.
- wrong_protein_theme (theme detection): the system has assigned the wrong (protein) theme to the event.
- wrong_event_theme (theme detection): the system has assigned the wrong (event) theme to the event.
- inexistent_theme2 (theme detection): the system has assigned a second theme to the binding event when there is no site associated to the event.
- no_theme2_found (theme detection): the system has not assigned a second theme to the binding event when it does exist.
- event_not_theme (theme detection): the system has assigned an event as the theme when it should have assigned a protein.
- protein_not_theme (theme detection): the system has assigned a protein as the theme when it should have assigned an event.
- diff_sentence_theme (different sentences): the event trigger and the theme are located in different sentences of the document.
- no_cause_found (cause detection): the system has not assigned a cause to the event when it does exist.
- inexistent_cause (cause detection): the system has assigned a cause to the event when there is no cause associated to it.
- inexistent_site (site/location detection): the system has assigned a site to the event when there is no site associated to it.
- no_site_found (site/location detection): the system has not assigned a site to the event when it does exist.
- no_atloc_found (site/location detection): the system has not assigned a location (AtLoc argument) to the localization event when it does exist.
- inexistent_toloc (site/location detection): the system has assigned a location (ToLoc argument) to the event when there is no location associated to it.
- no_toloc_found (site/location detection): the system has not assigned a location (ToLoc argument) to the localization event when it does exist.

The system which has been described in this section and together with the one of the section 4.3.2 has been integrated into the U-Compare Event Server and it is described in F.2 in Appendix.

	CAT	EXP	PHO	TRA	LOC	BIN	REG	POS	NEG
wrong_protein_theme	9(60,0%)	78(42,2%)	18(54,5%)	14(18,7%)	15(40,5%)	30(30,3%)	11(8,6%)	17(4,3%)	3(2,0%)
inexistent_site	-	-	2(6,1%)	-	-	-	-	1(0,3%)	-
no_site_found	-	-	2(6,1%)	-	-	1(1,0%)	2(1,6%)	-	-
composed_token_event	-	6(3,2%)	2(6,1%)	7(9,3%)	-	1(1,0%)	5(3,9%)	20(5,1%)	5(3,3%)
wrong_type_event	-	19(10,3%)	-	6(8,0%)	4(10,8%)	19(19,2%)	15(11,7%)	12(3,0%)	2(1,3%)
no_cause_found	-	-	-	-	-	-	-	8(2,0%)	1(0,7%)
inexistent_cause	-	-	-	-	-	-	4(3,1%)	12(3,0%)	3(2,0%)
wrong_event_theme	-	-	-	-	-	-	9(7,0%)	56(14,2%)	34(22,2%)
inexistent_theme2	-	-	-	-	-	3(3,0%)	-	-	-
inexistent_toloc	-	-	-	-	1(2,7%)	-	-	-	-
no_theme2_found	-	-	-	-	-	8(8,1%)	-	-	-
token_no_event	6(40,0%)	82(44,3%)	9(27,3%)	48(64,0%)	14(37,8%)	37(37,4%)	64(50,0%)	177(44,9%)	86(56,2%)
event_not_theme	-	-	-	-	-	-	10(7,8%)	27(6,9%)	8(5,2%)
protein_not_theme	-	-	-	-	-	-	8(6,2%)	64(16,2%)	11(7,2%)
no_toloc_found	-	-	-	-	3(8,1%)	-	-	-	-

Table 4.15: Details on the error analysis for the false positives.
For each type of error, the number of instances that have been found and the percentage that it represents for each type of event are shown. Therefore, for each column, the percentages sum approximately 100%. Abbreviation used for the events are the following: “EXP” for gene expression, “TRA” for transcription, “CAT” from protein catabolism, “PHO” for phsophorylation,”LOC” for localization, “BIN” for binding, “REG” for regulation, “POS” for positive regulation and “NEG” for negative regulation.

	CAT	EXP	PHO	TRA	LOC	BIN	REG	POS	NEG
wrong_protein_theme	2(66,7%)	12(14,5%)	4(28,6%)	-	2(9,1%)	53(25,6%)	9(6,2%)	12(2,5%)	8(5,1%)
inexistent_site	-	-	1(7,1%)	-	-	1(0,5%)	1(0,7%)	-	-
no_site_found	-	-	2(14,3%)	-	-	1(0,5%)	-	2(0,4%)	-
composed_token_event	-	-	-	1(3,4%)	1(4,5%)	1(0,5%)	4(2,7%)	15(3,1%)	4(2,5%)
wrong_type_event	-	9(10,8%)	-	9(31,0%)	1(4,5%)	6(2,9%)	3(2,1%)	50(10,4%)	4(2,5%)
argument_no_event	1(33,3%)	46(55,4%)	7(50%)	18(62%)	11(50%)	104(50%)	105(71,9%)	268(55,6%)	101(63,9%)
no_cause_found	-	-	-	-	-	-	2(1,4%)	15(3,1%)	3(1,9%)
inexistent_cause	-	-	-	-	-	-	2(1,4%)	5(1,0%)	-
wrong_event_theme	-	-	-	-	-	-	3(2,1%)	41(8,5%)	18(11,4%)
inexistent_theme2	-	-	-	-	-	16(7,7%)	-	-	-
no_case_found	-	10(12,0%)	-	1(3,4%)	2(9,1%)	11(5,3%)	3(2,1%)	27(5,6%)	5(3,2%)
trigger_no_event	-	3(3,6%)	-	-	-	4(1,9%)	1(0,7%)	12(2,5%)	8(5,1%)
no_atloc_found	-	-	-	-	1(4,5%)	-	-	-	-
no_theme2_found	-	-	-	-	-	5(2,4%)	-	-	-
event_not_theme	-	-	-	-	-	-	5(3,4%)	29(6,0%)	4(2,5%)
diff_sentence_theme	-	3(3,6%)	-	-	-	5(2,4%)	3(2,1%)	-	-
protein_not_theme	-	-	-	-	-	-	5(3,4%)	6(1,2%)	3(1,9%)
no_toloc_found	-	-	-	-	4(18,2%)	-	-	-	-

Table 4.16: Details on the error analysis for the false negatives.
For type of error, the number of instances that have been found and the percentage that it represents for each type of event are shown. Therefore, for each column, the percentages sum approximately 100%. Abbreviation used for the events are the following: “EXP” for gene expression, “TRA” for transcription, “CAT” from protein catabolism, “PHO” for phsophorylation,”LOC” for localization, “BIN” for binding, “REG” for regulation, “POS” for positive regulation and “NEG” for negative regulation.

4.5 Extraction of Disease and Treatment Relationships

In this section, we apply once more our general methodology for case-based reasoning (cf. 4.2) for the extraction of biomedical relationships. This time, we have decided to test it with a much simpler problem, the extraction of relationships between diseases and treatments. Our motivation here was to try the methodology with another corpus, besides the BioNL'09 Event Extraction Shared Task, the one which the methodology was developed for. For this purpose, we used the BioText corpus (cf. B.5).

As discussed before, the BioText corpus is not as complex as the biological events. Here, the entities which participate in the relationships are given. Also, there is only one relationship per sentence, which is always composed of one disease and one treatment. Also, there is no need to identify the relationship inside the sentence, as the BioText corpus is annotated by sentence-level. However, the relationships must be classified into the following classes: "PREVENT", "SIDE_EFF", "VAGUE", "TREAT_ FOR_DIS" and "TREAT_NO_FOR_DIS".

Only some few changes in the general algorithm were required to take into account the particularities of the domain. We only had to decide which features were more appropriate to represent the disease-treatment context and to implement the generation of those contexts, which comprises only one instance of each type of entity, one disease and one treatment. A case representative of a disease-treatment context is composed of the features listed in Table 4.17.

Element	Features	Type	
Context	Relationship	Nominal	**?**
	Entities of the context	Nominal	✓
Context Items	Part-of-speech tag	Nominal	×
	Type of entity	Nominal	×
	Role	Nominal	×
	Lemma	Nominal	×

Table 4.17: Features of the disease-treatment extraction classifier.
Features marked with an "x" are the ones considered for the corresponding element, the one marked with a "check" is mandatory and the feature-problem is identified by a question mark.

Regarding the features in Table 4.17, the relationship feature is the classification of the disease-treatment relationship. The value for this feature is given for the training dataset and unknown for the development dataset. The values it takes are: "PREVENT", "SIDE_EFF", "VAGUE", "TREAT_ FOR_DIS" and "TREAT_NO_FOR_DIS". The second feature at context level is the one representing the distinct types of entities composing the context. Because the disease-treatment relationship is simple, the types of entities involved are always the same (one disease and one treatment), but as this feature takes into account the

order in which they appear in the text, it may take two values ("DIS,TREAT" and "TREAT,DIS"); and, as it is a mandatory feature, it may be helpful as a first filtering of the cases.

The features for the context items includes the part-of-speech tag of the corresponding token, which was obtained with the Stanford parser (cf. C.4), the type of entity ("DIS", "TREAT" or "null"), the role of the item in the context ("DIS", "TREAT" or "null") and the lemma of the token, which is given by the Dragon toolkit (cf. C.3).

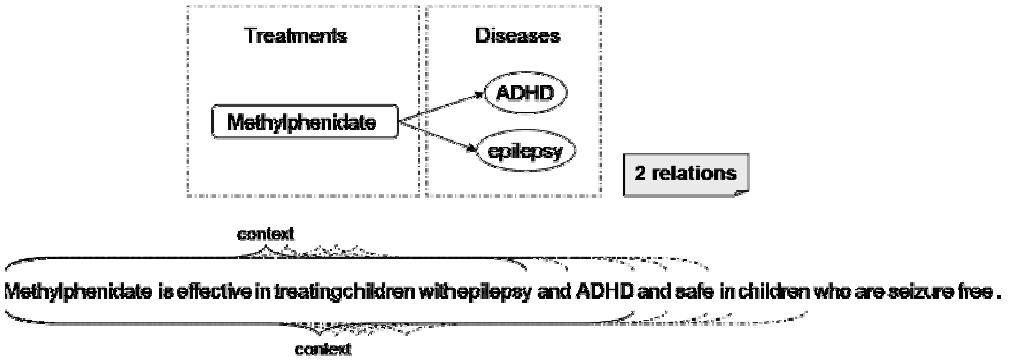


Figure 4.20: Contexts that have been generated according to the disease and treatment bags of entities of Figure 2A.
The length of the context may be delimited by the entities only (plain line) or by setting the number of tokens that come before and after them (dashed lines). The number of contexts to be generated from the bags of entities is always the product of the number of entities present in each of them.

There is a distinction between the type of entity and the role feature. For instance, in one of the contexts presented in Figure 4.20, the one in plain line that goes from the boundaries "Methylphenidate" and "ADHD" (below the sentence), the token "epilepsy" is of type "DIS" but takes no part in the context, i.e., its role is "null". Also, we decided to consider the lemma feature only for those tokens which are not roles of the context, as this feature's utility lies in identifying the classification of the relationship, whether it is vague or prevention, for example. The lemma of the named entities (disease and treatment) has not much influence on this decision.

We performed a 4-fold cross validation with the 964 sentences annotated with the disease and treatment entities (PREVENT, SIDE_EFF, VAGUE, TREAT_FOR_DIS or TREAT_NO_FOR_DIS). In each case, 75% of the sentences were used for training and 25% for testing. We carried out experiments for the four cross validation datasets and for a window from 0 to 10 for the tokens surrounding the context boundaries. The number of sentences in each of the

training and testing datasets is shown in Table 4.18. The dataset have been used in our experiments may be downloaded from the Moara project homepage[13].

Datasets / Relationships	Training				Testing			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
PREVENT	48	47	47	47	15	16	16	16
SIDE_EFF	23	23	22	22	7	7	8	8
VAGUE	27	28	28	28	10	9	9	9
TREAT_FOR_DIS	622	622	623	623	208	208	207	207
TREAT_NO_FOR_DIS	3	3	3	3	1	1	1	1
Total	723	723	723	723	241	241	241	241

Table 4.18: Distribution of the BioText corpus in the 4-fold cross-validation.
The number of sentences for the training and testing datasets for each fold of the cross-valdation and each of the categories is shown.

We selected as best results the higher micro-average and macro-average F-measures (cf. 3.5). The higher micro-average F-measure is calculated from the sum of the true positives across all classes. The higher macro-average F-measure is calculated as the average of the F-measure obtained by each class. Table 4.19 summarizes our best results, detailed by class. We have carried out experiments using various sizes for the surrounding window of tokens. Detailed results for each of these experiments are showed in Table 4.20 at the end of this section (in page 121). The best micro-average F-measure was obtained with a window [1,1] while the best macro-average result corresponded to a window [3,3].

From Table 4.19, the "TREAT_FOR _NO_DIS" and "SIDE_EFF" classes are certainly the harder ones to recognize. But this is mostly because the corpus is highly unbalanced, and few annotated sentences are available for these classes. However, the "SIDE_EFF" is surely much harder to predict than the "VAGUE" class, for instance, as the latter performs much better even when only a few training sentences are available.

No satisfactory comparison is possible between our results and those previously published. (Rosario and Hearst 2004) performed experiments using relevant and irrelevant documents. They call those sentences in which diseases or treatments are annotated 'relevant', so the classes "DISONLY" and "TREATONLY" are included in this dataset (cf. B.5). The 'irrelevant' sentences are those that are not annotated with any type of entity, i.e., those in the class "NONE". As showed in section B.5, we did not consider in our experiments those sentences that are annotated with only one entity or none. Also, (Rosario and Hearst 2004) performed experiments in which the disease and treatment are not given. When considering only relevant sentences, which include the "DISONLY" and "TREATONLY" classes, their best results had an accuracy of 92.5 with neural networks.

[13] http://moara.dacya.ucm.es/download/resources/biotext_cross_validation.zip

	Best Micro F-measure	**Best Macro F-measure**	**Higher F-measure**	**Training sentences**
PREVENT	54.5	51.6	56.3	48
SIDE_EFF	0.0	14.3	36.4	23
TREAT_NO_FOR_DIS	0.0	0.0	0.0	3
VAGUE	41.7	44.4	47.6	27
TREAT_FOR_DIS	91.0	90.5	91.3	622
All classes	**83.8**	**82.9**	**-**	**723**

Table 4.19: Results for the BioText corpus.
Results for the BioText corpus when only relevant documents annotated with both disease and treatment entities are considered. The macro-average F-measure is shown inside the parenthesis. The higher F-measure obtained for each class irrespective of cross-validation dataset or size of window is also shown.

Regarding the comparison with our results, the values for the F-measure and the accuracy are equivalent because we consider only positive (relevant) instances, although, as discussed above, distinct evaluation datasets were used. (Bundschus, Dejori et al. 2008) have also reported results using the BioText disease-treatment corpus. Their experiments took into account whether the named entities are or are not given, and they obtained accuracies of 96.9 and 79.5, respectively, when considering relevant and irrelevant sentences and using cascaded Conditional Random Fields.

Class	Dataset	Window of surrounding tokens [-x,+x]										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
All classes	**1**	83.40	**83.82**	82.57	82.99	82.99	83.40	82.99	81.74	81.33	80.50	80.91
	2	80.08	82.16	79.67	81.74	81.33	80.08	80.91	80.50	79.67	82.16	81.74
	3	78.01	83.40	82.99	80.91	79.67	76.76	78.01	78.42	79.25	79.25	79.25
	4	76.76	80.08	80.50	81.33	82.57	81.74	82.99	80.91	80.91	81.33	81.74
PREVENT	**1**	50.00	54.55	**56.25**	51.61	51.61	40.00	43.75	50.00	41.38	35.71	38.46
	2	27.59	31.25	14.29	22.22	20.00	13.33	20.69	20.69	18.75	37.50	32.26
	3	13.33	25.00	22.22	21.43	13.33	12.12	6.45	13.33	12.50	12.12	12.12
	4	35.90	31.25	15.38	23.08	25.00	16.67	24.00	15.38	8.00	14.81	15.38
SIDE_EFF	**1**	0.00	0.00	15.38	14.29	15.38	16.67	18.18	14.29	14.29	14.29	15.38
	2	26.67	28.57	30.77	26.67	28.57	30.77	18.18	20.00	**36,36**	**36,36**	**36,36**
	3	25.00	30.77	30.77	18.18	18.18	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	15.38	18.18	18.18	20.00	16.67	16.67	16.67	**36,36**	**36,36**	**36,36**
TREAT_NO_FOR_DIS	**1**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
VAGUE	**1**	**47.62**	41.67	37.04	44.44	35.71	44.44	41.38	35.71	34.48	33.33	32.26
	2	0.00	25.00	0.00	26.67	25.00	23.53	21.05	21.05	21.05	21.05	21.05
	3	36.36	40.00	42.11	28.57	33.33	34.78	36.36	33.33	36.36	38.10	38.10
	4	0.00	0.00	11.76	11.11	20.00	30.00	31.58	22.22	19.05	20.00	20.00
TREAT_FOR_DIS	**1**	91.04	91.04	89.98	90.46	90.95	**91.26**	90.95	90.15	90.20	89.71	89.76
	2	88.84	90.21	88.47	89.62	89.79	88.84	89.57	88.89	88.31	89.74	89.52
	3	87.38	90.78	90.74	89.52	88.94	86.68	87.77	87.98	88.73	88.73	88.73
	4	86.83	89.26	89.20	89.67	90.40	89.88	90.82	89.41	89.62	89.83	90.09

Table 4.20: Details on the experiments with different window lengths.
Results are presented for using 4-fold cross-validation (dataset column) each type of relationship and for a variable window of tokens.

4.6 Recognition of Gene and Protein Mentions

In this section, we will present our methodology for the extraction of genes and protein from scientific literature. We propose the use of an ensemble of taggers, which includes a tagger of our own based on the case-based reasoning approach (Aamodt and Plaza 1994). Here, our CBR approach is slight different to the one proposed in section 4.2, and it will be described in details in this section.

It consists of two parts: training and testing. In a first step several cases of the two classes (gene mention or not) are stored in two bases (one for the known and one for the unknown cases). During the testing step, the system searches these bases for the case most similar to the problem and finally a classification decision is given by the class of the case selected as the most similar. For the training and testing step, the BioCreative 2 Gene Mention training and testing datasets (cf. B.1) have been used, which consists of 15,000 and 5,000 sentences, respectively.

During this first step, the training dataset was split in 10 subsets in order to perform a 10-fold cross validation test. The sentences were extracted from the training documents and the tokens are separated using spaces and punctuations as separators (including parenthesis, brackets, symbols, etc.). These were the tokens used to construct the two case bases, one for known cases and the other for unknown cases, as proposed for the part-of-speech tagging problem in (Daelemans, Zavrel et al. 1996).

The known cases are used by the system to classify tokens that are not new, i.e. tokens that have appeared in the training documents. The attributes used to represent a known case are the following:

- the token itself;
- the category of the token (if it is a gene mention or not);
- and the category of the preceding token (if it is a gene mention or not).

Each token represents a single case, and repetition of cases with exactly the same attributes is not allowed. In order to account for repetitions, the frequency of the case is incremented to indicate the number of times that it appears in the training dataset. The frequency of a case will be taken into account in the search procedure, the higher the frequency of a case, the higher the probability that the case may be chosen.

Tokens composed of only numbers and/or Greek letters are not added to the case base. Token are always saved exactly as they appear in the text, in upper case or lower case. Additionally, cases that do not represent gene mention are also saved in lower case because we suppose that the case of the letter is not so important for recognizing non gene mention tokens. In addition, parts of a gene mention are also added to the case base and these tokens are divided according to some symbols, such as ‘/’, ‘-’, ‘+’, etc. For example, with the token “Uga35p/Dal81p/DurLp”, the system first includes it as a

whole and then separates it according to the slashes and saves the tokens "Uga35p", "Dal81p" and "DurLp" as well.

The unknown base is used to classify tokens that were not present in the training documents. The unknown cases are built over the same training data used for the known cases. Instead of saving the token itself, a shape of the token is kept in order to allow the system to classify unknown tokens by looking for cases with similar shape. Therefore, as in the known cases, the attributes that have been used to represent the unknown cases are the following:

- the shape of the token,
- the category of the token (if it is a gene mention or not);
- and the category of the preceding token (if it is a gene mention or not).

The system saves these attributes for each token in the sentence as an unknown case. As with known cases, no repetition is allowed and instead the frequency of the case is incremented.

The shape of the token is given by its transformation into a set of symbols according to the type of character found: "A" for any upper case letter; "a" for any lower case letter; "1" for any number; "p" for any token in a stopwords list (cf. Appendix E.1); "g" for a Greek letter; "$" for identifying 3-letter-prefixes and 4-letter-suffixes in a token. For example, "Dorsal" is represented by "Aa", "Bmp4" by "Aa1", "the" by "p", "cGKI(alpha)" by "aAAA(g)", "patterning" by "pat$a" ('$' separates the 3-letter prefix) and "activity" by "a$vity" ('$' separates the 4-letters suffix). The symbol that represents an uppercase letter ("A") can be repeated to take into account the number of letters in an acronym, as shown in the example above. However, the lowercase symbol ("a") is not repeated; suffixes and prefixes are considered instead. These are automatically extracted from each token by considering the last 4 letters and first 3 letters, respectively; they do not come from a predefined list of common suffixes and prefixes.

In the construction of cases, the training documents are read twice, one in the forward direction (from left to right), and one in the backward direction (from right to left). This is done to allow a more variety of cases due to the fact that the decision of classifying a token as a gene mention may be influenced by its preceding and/or following tokens.

The known and unknown cases saved during the forward reading are used for forward classification only, as well as the backward cases are useful only to the backward classification. However, a decision carried out during one of the reading directions may be used by the other direction. For example, when starting by the forward directions, the decision made for a token (being a gene mention or not) may be used as the category of the preceding token. This procedure may help recognizing mentions composed by more

than one token that would not have been totally recognized when considering one of the reading procedures only.

The training procedure consists of moving a sliding window from the start to the end of the text in both directions. For each token, the system keeps track of the category of the preceding token (false at the beginning), gets the category of the actual token, according to the gold standard provided with the dataset, and saves it both in the known and unknown case base.

The classifier has also been trained with additional corpora in order to be able to better extract mentions from different organisms. These extra corpora belong to the gene normalization training datasets for the BioCreative task 1B (cf. B.1) corresponding to yeast, mouse and fly. Since these training documents were only annotated with the identifiers of the genes/proteins and not with the mentions themselves, these documents were first annotated by performing, for each organism, an exact matching of the synonyms of the dictionary with the tokens of the training documents. Consequently, no approximated matching of mentions and synonyms were able to be extracted, the taggers were trained with the exact matches exclusively.

The consideration of the mentions obtained using only an exact matching has some impacts in the results. Ideally, it should be very useful to have all the mentions present in the document, by an approximated matching or by manually annotating them. Unfortunately, an approximated matching with a whole dataset of documents is time consuming and a manually annotation of the mentions would need some experts in the organisms under consideration. Anyway, this process has improved the performance of the gene normalization step, especially for the fly dataset due maybe to the lack of examples for this organism's nomenclature.

Therefore, the CBR-Tagger classifier has been trained with the training set of documents made available during the BioCreative 2 Gene Mention task as well as the gene normalization datasets for the BioCreative task 1B for the yeast, mouse and fly. These training datasets will be referred to hereafter as CbrBC2, CbrBC2y, CbrBC2m, CbrBC2f and CbrBC2ymf, depending if they are composed by the BioCreative 2 Gene Mention task corpus alone or combined with the BioCreative task 1B corpus for the yeast, mouse, fly or all three, respectively.

During the testing step, the system searches the known and unknown bases for the case most similar to the problem and a classification decision is given by the class of the case selected as being most similar. The classification procedure works in a similar way to the construction of cases. The text is tokenized and a sliding window is applied in the forward direction and then in the backward direction. In each case, the system keeps track of the category of the preceding token (false at the beginning), gets the shape of the token (according to the symbols described above) and attempts to find a case most

similar to it in the base. If more than one case if found, the one with the higher frequency is chosen.

The search procedure is separated into two parts, one for the known cases and another for unknown cases. In this search strategy, priority is given to the known cases. For known cases, the token is saved exactly as it appeared in the training documents, and the classification is more precise than using unknown cases. The system also separates the token into parts in order to classify them individually. As pointed out in section 4.1, the CBR life cycle (Aamodt and Plaza 1994) allows the re-training of the system with the experience learnt from retrieved cases. However, the CBR-Tagger does not include this step in its methodology.

The importance of considering both directions for reading the text lies in the fact that sometimes the tokens are more easily recognized as a gene mention in only one of the two directions, especially for mentions composed of more than one token. For example, for the mention "cka1 delta cka2-8" from the BioCreative 2 Gene Mention task dataset, the tokens "cka1" and "cka2" are easily recognized in any direction, forward or backward, as they have always appeared in the training documents as gene mentions. By looking at the cases in the bases, "delta" would never be recognized as a gene mention in the forward direction, as in most of the cases in the training documents, "delta" appeared as a not gene mention token, no matter if a gene mention was preceding it or not, in the forward direction. However, when looking the cases for the backward direction, if a gene mention comes before it (i.e, after it in the text), as it is the case of "cka2" in "cka1 delta cka2-8", "delta" is easily classified as a gene mention too.

Correctly classified			Incorrectly classified	
Kv1.	rMsERK1	TAp63gamma	4Fe4S	P156KKIKP161
GRB2.	cdc42W97R	p46kDaPax	YBa2Cu	pSP64E6E7
1935UF	ERCC3Dm	HNF4alpha7	C2GnT	W89RKRRY94
2Apro	D12S2293	YlRim101p	6m141	VVDeltaE3L
B19p6	APprog	tom1C3235A	Tc99MDP	15AcDON
CRAlBP	PFP9a20	D10S1789	PC8SRaw	3AcDON
D8S520	alpha2C4	PFP9a20	100gamma	90dBnHL
p130CAS	P450IIE1		serine49	GGTCTnnnAGACC
D3F15S2	p110RB1		Lys381Lys382Leu383Met384Phe385	

Table 4.21: Correct and incorrectly tokens classified as default as gene/protein mentions. List of the correct and incorrectly tokens classified as default as gene/protein mentions for the BioCreative 2 Gene Mention task testing dataset.

If no best known or unknown case is found in the search procedure, the token is classified as a gene mention by default. This decision is due to the fact that if the token and its shape are both strange to the system, it must have a very special shape and the possibilities that it may be a gene mention is high, as the gene/protein nomenclature usually includes sequences of letters, numbers and symbols (such as hyphens, slashes, parenthesis, etc.). Some examples of this case are the following tokens: "ERCC3Dm",

"cdc42W97R", "IL7Ralpha" and "p46kDaPax". Our experiments have shown than this hypothesis is true in about 61% of the decisions. An analysis of the correct and incorrect mentions that fall in this case has reported no explicit pattern that could be considered by the system. These correct and incorrect mentions are presented in Table 4.21.

After the identification of the best case for each token, post-processing procedures are executed in order to check the boundaries, especially important for mentions composed of more than one token, as well as abbreviations and corresponding full names. These post-processing steps check small gaps (of one or two tokens) between two identified mentions and decide if they should be also recognized as part of the mention or not. The abbreviation step, which is part of the post-processing phase, checks if a complete form of the mention appears before the abbreviation by associating each of its character with a token that comes before, taking into account their order in the abbreviation. For example, in the case of the mention "mixed lineage kinase 3 (MLK3)", the system was able to identify the tokens "kinase" (known case, backward reading), "3" (unknown case, backward reading) and "MLK3" (unknown case, forward reading). The rest of the tokens "mixed", "lineage" and the parenthesis were only recognized by the post-processing steps.

Training set	Recall	Precision	F-Measure
CbrBC2	64.11	76.01	69.56
CbrBC2y	42.90	80.98	56.08
CbrBC2m	29.14	76.08	42.14
CbrBC2f	51.05	73.66	60.30
CbrBC2ymf	24.53	77.00	37.21
Best BioCreative	85.97	88.48	87.21
BANNER	82.78	87.18	84.92
ABNER	51.49	86.93	64.68

Table 4.22: Results for the gene/protein recognition.
Results are shown for the five training datasets under consideration here (CbrBC2, CbrBC2y, CbrBC2m, CbrBC2f and CbrBC2ymf) when training our methodology. We also present the best result on the BioCreative 2 Gene Mention challenge and using BANNER and ABNER taggers. The results are for the testing dataset of the BioCreative 2 Gene Mention.

An initial version of this methodlogy (Neves 2007) has participated in the BioCreative 2 Gene Mention task (Smith, Tanabe et al. 2008). One of the differences between these two approaches is that the one described here reads the documents in two directions. Also, some improvements have been included to the methodology, such as changes in the shape of the unknown cases, including the consideration of suffixes and prefixes, as well as the post processing steps in order to take into account the boundaries of the mention and abbreviations.

We have evaluated our system with the 5,000 documents that compose the BioCreative 2 Gene Mention testing dataset (cf B.1). In Table 4.22, results are presented according to the five datasets used for training the tagger: CbrBC2, CbrBC2y, CbrBC2m,

CbrBC2f and CbrBC2ymf. The last two lines of the table present the best results recently published using the BioCreative 2 Gene Mention dataset as well as BANNER and ABNER results when trained with the latter training corpus.

The results showed in Table 4.22 confirm that the CbrBC2 is the best dataset for training CBR-Tagger to the gene mention recognition problem. However, results presented in section 5.2 for gene/protein normalization show that in some cases, depending of the organism in consideration, a tagger trained with specific documents may improve the recall and F-Measure for this task.

It is understandable that the BC2 Gene Mention results are better when trained with the BC2 corpus only. The mistakes made by the system when training the CBR-Tagger with the documents that belonged to the BioCreative task 1B corpus (provided for the gene/protein normalization task) were usually for mentions that belonged to other organisms. The BioCreative task 1B corpus consists of three independent corpora each one automatically annotated for only one of the organisms (yeast, mouse or fly). However, it may contain mentions from other organisms which have not been annotated. For example, mentions of the human may appear in the mouse corpus, but they would be learned as negative cases by our tagger because only those mentions of the mouse (the ones annotated) would be learned as positive. Some examples of these mistakes are the tokens: “immunoglobulin”, “EprexR” and “thymidine kinase”.

Cases / Reading	**Forward**			**Backward**			**Forward + Backward**		
	P	R	FM	P	R	FM	P	R	FM
Known cases	29.23	69.36	41.13	27.59	66.91	39.07	30.91	74.71	43.73
Unknown cases	21.68	29.33	24.93	23.00	31.24	26.49	24.97	35.02	29.15
Known and unknown cases	73.58	61.80	67.18	70.47	58.92	64.18	72.69	67.43	**69.96**
+ post-processing	76.15	60.21	67.25	72.59	55.90	63.16	76.01	64.11	**69.56**

Table 4.23: Performance of the gene/protein tagger according to its configuration.
Results are presented for both forward and backwards direction, when considering the know and unknown bases of cases and when adding the pros-processing step.

On the other side, the BioCreative Gene Mention corpus was annotated by experts and includes mentions from any organisms. When training the system with this corpus, it might be able to recognize mentions from any organism reasonably, while by training it with the normalization corpus, the tagger may be biased to the mentions of the corresponding organism only. However, it may still be helpful if the user is interested in a tagger more specific to a given organism.

In the case of reading the text in the forward and backward direction, experiments have shown that both directions performs similar, being the forward reading slightly better. However, using both of them together may improve the F-Measure in about two points more than using only the forward direction. Detailed results of the experiments carried

out using only forward or backwards reading is presented in Table 4.23. In this table, results are also presented when comparing the use only of known or unknown cases, as well as the influence of the post-processing step.

Although the results presented for the gene/protein mention extraction are below the best BioCreative results, this task is considered as a preceding step for gene/protein normalization, and the improvement of this normalization is the main goal of a tagger. Regarding the errors, false negatives in the gene/protein recognition step are not always a problem since the normalization task may be performed successfully if others (different) mentions of the same gene/protein have been able to be extracted from the text. Also, when combining CBR-Tagger with other taggers (such as ABNER or BANNER), our experiments (cf. Table 5.6) showed that it improves the final results.

This approach which has been presented in this section for the gene/protein recognition has been used for the development of the CBR-Tagger system (Neves, Carazo et al. 2010) as part of the Moara project[14]. Detailed functionalities of the system and examples of use are described in details in section F.1, in the Appendix of this thesis.

[14] http://moara.dacya.ucm.es

4.7 Summary of the Chapter

In this chapter we have described the methodologies we propose for the entities and relationships extraction. They were all based based on the case-based reasoning (CBR) approach which was introduced in section 4.1.

In section 4.2 we explained in details the general approach that we propose for using CBR for information extraction. The training and the testing steps are described. The representation of the cases is one of the most important points in the case-based reasoning approach and we proposed two alternatives for the context (a sup-part of a sentence) under study: context based on a window of tokens (cf. 4.2.1.1) and context defined by some predefined tokens (cf. 4.2.1.2). We also presented details on how the contexts (and the respective cases) are automatically generated in our methodology. As CBR solves a problem by searching in the base for its most similar case, we also described in details the two types of similarities we use: exactly matching (cf. 4.2.3.1) and global alignment of the features (cf. 4.2.3.2).

We have evaluated our methodology to four different biomedical problems, two related to named-entity recognition (gene and proteins in section 4.6 and for the biological event triggers in section 4.3) and two for the relationships extraction (biological events in section 4.4 and relationships between disease and treatments in section 4.5). When necessary, details are highlighted regarding the implementation of the general methodology for each task, such as the features we have used, the representation of the contexts and the type of similarity used to compare the cases.

CHAPTER 5 NORMALIZATION OF GENE AND PROTEIN MENTIONS

The gene/protein normalization procedure is the problem of associating an identifier in a dictionary of synonyms (which may be specific of a given organism) to a potential gene/protein mention previously recognized in the text. This chapter will describe the methodologies we propose to solve this difficult task. In particular we present three approaches: exact matching based on edited synonyms (cf. 5.2), approximated matching based on trie and global alignment (cf. 5.3) and approximate matching based on machine learning (cf. 5.4). For all of them, we consider that the gene/protein mentions have already been extracted using any available tagger, such as the one developed in this thesis and previously discussed in section 4.6.

5.1 Construction of the Dictionary of Synonyms

For all the methodologies proposed here, the dictionary of synonyms used for the construction of the normalization system is based on the ones provided by the Bio-Creative competitions, which contains 14,995 synonyms for yeast, 130,208 synonyms for mouse, 116,744 synonyms for fly (cf. B.2) and 203,077 synonyms for human (cf. B.3). Some operations might have been necessary to be carried out in the dictionary of synonyms according to the matching methodology proposed, such as editing operations (cf. 5.2) or ordering it according to a determined data structure, such as for the trie (cf. 5.3), which will be described in the respective sections of this work.

5.2 Exact matching

The exact matching we propose consists of checking the coincidence between a mention and the synonyms in the dictionaries. However, the matching is carried out not with the original mention and synonyms, but with variants for both of them. The mention and the dictionary of synonyms are previously pre-processed by applying some editing operations. This editing operation is carried out only once to the dictionary of synonyms during the development of the system. However, it has to be performed for each mention during the normalization procedure. These editing operations are carried out equally to any organism.

First of all, the tokens are converted to lower case and its tokens separated according to some boundaries such as symbols (e.g., plus signal, etc.) punctuations (e.g., commas, colon, semi-colon, parenthesis, brackets, bars, hyphen, underscore, etc), Greek letters and numbers. These subparts are then ordered alphabetically, as proposed in (Liu, Wu et al. 2004), in order to avoid mismatching due to different ordering of the same tokens. For example, the gene “N-myristoyl transferase” (FlyBase identifier FBgn0020392) may sometimes be written as “N myristoyl transferase” (no hyphen). By converting these synonyms to lower case and ordering their parts, the resulting token “myristoyl n transferase” is much more flexible and would be matched to any of the two synonyms, “N-myristoyl transferase” and “N myristoyl transferase”, with or without the hyphen.

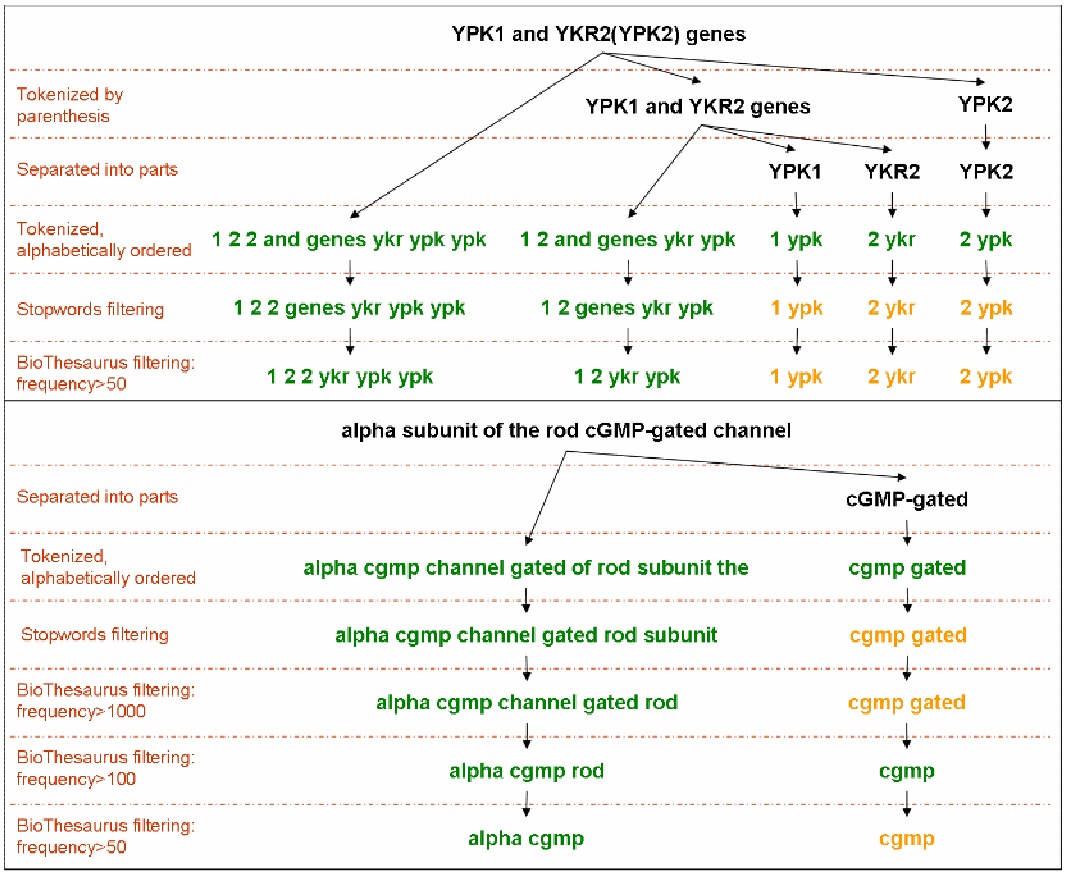


Figure 5.1: Editing procedures for the generation of mention and synonym variations. Two examples of the editing procedures are shown in detail. The non-repeated variations that are returned by the system are presented in green and the repeated variations are shown in orange. Only those procedures that result in a change to the examples are shown.

The system also performs a cleaning of the mention (or synonym) in order to remove parts of it which coincides with the biomedical terms in BioThesaurus[15] list, stopwords[16] (cf. Appendix E.1) or organism's names from NCBI Entrez Taxonomy database[17]. This is especially helpful for mentions (or synonyms) composed of many tokens. For example, the mention "alpha subunit of the rod cGMP-gated channel" would be transformed to "cgmp phosphodiesterase rod" after the cleaning and ordering procedures.

The cleaning of the biomedical terms of the BioThesaurus list is accomplished gradually according to the frequency of the term in the list. For example, for the same mention "alpha subunit of the rod cGMP-gated channel", the frequencies of the terms "subunit", "gated" and "channel" are 4092, 152 and 794 in BioThesaurus, respectively. By making the cleaning process gradually, according to the terms whose frequencies are

[15] http://pir.georgetown.edu/pirwww/iprolink/protname.shtml
[16] http://www.unine.ch/info/clef/englishST.txt
[17] http://www.ncbi.nlm.nih.gov/sites/entrez?db=taxonomy

higher than 10, 1000 and 10000, the mention (or the synonym) would be saved as "cgmp phosphodiesterase rod", "cgmp channel gated phosphodiesterase rod" and "cgmp channel gated phosphodiesterase rod subunit", respectively. This procedure generates many variations of the original mention (or synonym). With such a procedure, we increase the possibility of finding an exact matching with no need of providing information specific of an organism.

Figure 5.1 illustrates the editing procedure for two examples: "YPK1 and YKR2 (YPK2) genes" and "alpha subunit of the rod cGMP-gated channel". The figure has been simplified in order to include only those steps which generate a new variation of the preceding text in each of the examples. Therefore, the filtering excluded BioThesaurus terms with frequencies higher than 10,000, 10 or zero.

	Yeast			**Mouse**			**Fly**			**Human**		
	P	R	FM	P	R	FM	P	R	FM	P	R	FM
Manual	**88.21**	88.89	88.55	52.00	72.69	60.63	55.79	58.30	57.02	40.08	74.53	52.13
Bio zero	87.08	**90.42**	88.72	60.39	73.12	66.15	**60.11**	49.33	54.19	**55.26**	70.63	**62.00**
Bio10	87.08	**90.42**	88.72	59.04	74.41	65.84	59.28	51.57	55.16	53.61	71.88	61.42
Bio50	87.08	**90.42**	88.72	55.30	73.98	63.29	56.11	55.61	55.86	48.71	73.91	58.72
Bio100	87.41	**90.42**	**88.89**	51.72	74.19	60.95	57.21	56.95	57.08	45.43	75.31	56.67
Bio1000	87.36	90.04	88.68	53.28	**75.05**	62.32	54.96	**59.64**	**57.20**	43.00	**76.25**	54.99
Bio10000	87.22	88.89	88.05	54.84	73.12	62.67	55.55	58.30	56.89	42.68	76.09	54.69
No filter	87.22	88.89	88.05	58.96	72.90	65.19	55.32	58.30	56.77	50.16	75.16	60.16
Hybrid	87.08	90.42	88.72	**65.16**	74.41	**69.48**	58.79	52.47	55.45	53.14	74.06	61.88

Table 5.1: Gradual filtering of the biomedical terms.
Results are shown for the BioCreative 1 (yeast, mouse, fly) and 2 (human) development datasets when using different thresholds (0, 10, 50, 100, 1000, 10000 and all of them) for filtering the biomedical terms from the BioThesaurus list.

Regarding the BioThesaurus, we consider the complete lexicon in our filtering step, i.e., the files identified as "BioMedical terms", "Chemical terms", "Macromolecules" ("enzymes", "single word names" and "general names"), "Common English" and "Single non-word tokens". We perform filtering for the terms identified as "gn" and "pr", as they indicate terms that refer to genes and proteins.

One-token mentions or synonyms were filtered using those biomedical terms with frequency higher than 10 in the BioThesaurus list, using the gradual cleaning procedure described above. Regarding the filtering of part of a token or a token from a multi-token mention, some experiments were carried out in order to select the best threshold according to the following configurations (cf. Table 5.1):

- Manual list: list constructed manually and composed by 40 terms;
- Bio zero: filtering all the term in the BioThesaurus list (3436 terms);
- Bio10: filtering the terms which frequency is higher than 10 (2223 terms);
- Bio50: filtering the terms which frequency is higher than 50 (701 terms);
- Bio100: filtering the terms which frequency is higher than 100 (378 terms);
- Bio1000: filtering the terms which frequency is higher than 1000 (38 terms);
- Bio10000: filtering the terms which frequency is higher than 10000 (8 terms);
- filtering no terms;
- a hybrid strategy in which the filtering of the whole mention (or synonym) is filtered by all the term with frequency higher than 10 (Bio10) and the part of the mention/synonym are filtered gradually starting from Bio zero until Bio10000.

	Yeast			**Mouse**			**Fly**			**Human**		
	P	R	FM	P	R	FM	P	R	FM	P	R	FM
Bio zero	**87.08**	**90.42**	**88.72**	**65.96**	73.33	69.45	**59.57**	50.22	54.50	**53.94**	72.81	**61.97**
Bio10	**87.08**	**90.42**	**88.72**	65.16	74.41	**69.48**	58.79	52.47	55.45	53.14	74.06	61.88
Bio50	**87.08**	**90.42**	**88.72**	63.50	74.84	68.71	55.36	55.61	55.48	47.87	**75.63**	58.63
Bio100	**87.08**	**90.42**	**88.72**	62.66	75.05	68.30	55.70	**56.95**	56.32	47.87	**75.63**	58.63
Bio1000	**87.08**	**90.42**	**88.72**	59.02	75.27	66.16	55.70	**56.95**	56.32	47.87	**75.63**	58.63
Bio10000	**87.08**	**90.42**	**88.72**	58.50	**75.48**	65.91	55.95	**56.95**	**56.44**	47.92	**75.63**	58.67

Table 5.2: Gradual filtering of the biomedical terms for the hybrid alternative.
Results are shown for the BioCreative 1 (yeast, mouse, fly) and 2 (human) development datasets when using different thresholds for filtering the biomedical terms from the BioThesaurus list in the hybrid configuration.

The terms that compose the manual list are the following: "dna", "rna", "kinase", "mrna", "kbp", "trna", "rrna", "element", "transcript", "factor", "cdna", "domain", "receptor", "homolog", "region", "chromosome", "product", "type", "growth", "subunit", "protein", "proteins", "molecule", "molecules", "peptide", "antigen", "mitochondrial", "s. cerevisiae", "saccharomyces cerevisiae", "yeast", "mouse", "mice", "human", "h. sapiens", "fly", "drosophila", "melanogaster", "drosophila melanogaster", "kb", "bp".

In regard to the hybrid configuration showed in Table 5.1, we have decided to use the "Bio10" terms for filtering the terms after carrying out the experiments listed in Table 5.2. Here, the biomedical terms that appear as part of the token are always filtered gradually while the whole mentions (or synonyms) are filtered according to the shown configuration.

The system also removes those mentions and synonyms composed of only one character, parts of them with less than four characters (after cleaning procedures), composed by no letters at all (i.e., only numbers or other symbols), or that coincides to Roman numeral, Greek letters, amino acids, stopwords (cf. Appendix E.1) and organism's names (from the NCBI Entrez Taxonomy database). Some comparative

experiments were also carried out in order to evaluate the methodology according to these pre-processing steps, as showed in Table 5.3.

	Yeast			Mouse			Fly			Human		
	P	**R**	**FM**	**P**	**R**	**FM**	**P**	**R**	**FM**	**P**	**R**	**FM**
Baseline	**91.4**	81.2	86.0	**69.9**	68.0	68.9	**60.1**	50.7	55.0	**65.8**	64.2	**65.0**
+ lower case	**91.4**	81.2	86.0	**69.9**	68.0	68.9	59.5	50.7	54.7	**65.8**	64.2	**65.0**
+ ordering	84.5	90.0	87.2	31.2	72.9	43.7	39.8	**54.3**	45.9	29.9	**77.0**	43.1
+ numbers and Greek letters	72.9	**90.8**	80.9	27.2	73.1	39.7	33.2	**54.3**	41.2	26.7	**77.0**	39.7
+ clean basic	88.1	**90.8**	**89.4**	57.4	72.7	64.1	56.8	**54.3**	**55.5**	52.7	72.0	60.9
+ BioThesaurus	87.1	90.4	88.7	65.2	**74.4**	**69.5**	58.8	52.5	55.5	53.1	74.1	61.9

Table 5.3: Comparison of the processing steps for the exact matching.
Results are presented for the BioCreative 1 (yeast, mouse, fly) and 2 (human) development datasets. The final configuration of the system uses all the features above listed. "P" stands for precision, "R" for recall and "FM" for F-measure.

Regarding the dictionary of synonyms, the number of resulting synonyms for each organism after the editing operations is presented in Table 5.4. These edited synonyms are the ones that are used in the matching procedures; the original ones are kept as reference only. We have made available a list of the pre-processed synonyms used in our flexible matching strategy in the download page[18] of the Moara Project website. In summary, the original lists were the ones described in section 5.1. The final list is the one obtained after some editing steps such as converting the synonyms to lower cases, tokenizing them by some symbols and gradually cleaning them according to BioThesaurus lexicon of biomedical terms.

List / Organism	**Yeast**	**Mouse**	**Fly**	**Human**
Original	14,995	130,208	116,744	203,077
Edited	15,111	209,359	113,950	289,707

Table 5.4: Size of the dictionaries of synonyms before and after the preprocesing.
The number of synonyms in the gene/protein normalization dictionary before and after the editing operations is this presented for each organism.

After all these preprocessing tasks, the surviving mentions (or synonyms) are ready to be presented to the matching procedure in order to decide their gene/protein identifiers. The matching strategy simply consists of performing an exact matching between the (edited) mentions extracted by the tagger and the (edited) synonyms of the dictionary.

[18] http://moara.dacya.ucm.es/download.html

This procedure cannot be considered as a perfect exact matching as both the mentions and the synonyms have been previously modified as described above. If more than one synonym is matched for a same mention, a disambiguation step is followed, as discussed in section 5.5.

Table 5.5 presents the results for the organisms under consideration here, i.e. yeast, mouse, fly and human, for the exact matching, along with the best results published recently for the same datasets. The datasets used for the evaluation are the ones from BioCreative task 1B for the yeast, mouse and fly (cf. B.2) and the BioCreative II Gene Normalization task (cf. B.3). The best results for yeast and fly were obtained using the BioCreative task 1B (Hirschman, Colosimo et al. 2005) and for mouse and human were obtained using GNAT (Hakenberg, Plake et al. 2008). Additionally, GENO (Wermter, Tomanek et al. 2009) reports an overall F-Measure performance of 86.4 over the BioCreative 2 test set. The results presented in this table for our normalization method uses an ensemble of three taggers (ABNER, BANNER and CBR-Tagger) for extraction the gene/protein mentions (described in details below) and a single disambiguation based on cosine similarity (cf. 5.5).

Organism	**Best results (BioCreative and GNAT)**			**Moara results**		
				Exact matching		
	Recall	**Precision**	**F-Measure**	**Recall**	**Precision**	**F-Measure**
Yeast	89.4	95.0	92.1	83.52	95.17	88.97
Mouse	91.6	72.6	81.0	77.57	65.83	71.22
Fly	80.0	83.1	81.5	69.76	59.12	63.58
Human	90.1	81.1	85.4	83.31	55.00	66.26

Table 5.5: Results for the exact matching for the gene/protein normalization task
Results are shown for the test corpora. Best results by organism for the gene/protein normalization task evaluated with the test corpora of the BioCreative 1 task 1B (yeast, mouse and fly) and BioCreative 2 Gene Normalization task (human). The results were produced using a mix of Abner, Banner and CBR-Tagger (CbrBC2ymf) and single disambiguation by cosine similarity multiplied by the number of common words.

We have also performed experiments using the gold standard gene/protein annotations, instead of extracting them with the proposed tagger. This methodology was carried out only for the human dataset, as the gold standard datasets for the other organism were not annotated with the gene/protein mentions, but only with their identifiers in the respective databases. This experiment resulted in a precision of 83.36, a recall of 75.94 and F-measure of 79.48 for the disambiguation strategy used for the results of Table 5.5. These results are considerably higher than the ones obtained when using the ensemble of taggers.

In order to check the influence of the taggers in the normalization problem, we have performed experiments using the many datasets we proposed for the training of the CBR-Tagger (cf. 4.6), as well as its combination with ABNER (cf. C.5) and BANNER

(cf. C.6) taggers. Table 5.6 shows the results for the fly dataset when performing exact matching. Experiments were accomplished using only one tagger and a mix of two or three of them. Three different configurations were tried for the CBR-Tagger according to the set of documents used in the training step: BioCreative 2 Gene Mention task only (CbrBC2), a combination with the BioCreative 1 task 1B for fly (CbrBC2f) and also for the three organisms corpora (yeast, mouse and fly) (CbrBC2ymf) available in the latter challenge.

Mix of Taggers		Yeast			Mouse			Fly			Human		
		P	R	FM	P	R	FM	P	R	FM	P	R	FM
Gold standard mentions		-	-	-	-	-	-	-	-	-	83.36	75.94	79.48
one tagger	**Abner**	90.32	75.10	82.01	63.62	64.30	63.96	54.47	30.04	38.73	57.99	64.06	60.88
	Banner	90.73	71.26	79.83	63.67	68.60	66.05	58.54	32.29	41.62	55.64	70.94	62.36
	CbrBC2	92.86	59.77	72.77	66.76	53.12	59.16	59.05	27.80	37.80	56.92	57.19	57.05
	CbrBC2y	94.18	68.20	79.11	-	-	-	-	-	-	-	-	-
	CbrBC2m	-	-	-	76.32	37.42	50.22	-	-	-	-	-	-
	CbrBC2f	-	-	-	-	-	-	71.72	46.64	56.52	-	-	-
	CbrBC2ymf	91.33	68.58	78.34	77.95	32.69	46.06	77.48	38.57	51.50	76.79	28.44	41.51
two taggers	**Abner + Banner**	89.74	80.46	84.85	59.68	71.61	65.10	53.90	37.22	44.03	52.96	72.66	61.26
	Abner + CbrBC2	89.82	77.78	83.37	60.67	70.32	65.14	53.16	37.67	44.09	53.29	70.94	60.86
	Abner + CbrBC2y	90.09	80.08	84.79	-	-	-	-	-	-	-	-	-
	Abner + CbrBC2m	-	-	-	63.28	69.68	66.33	-	-	-	-	-	-
	Abner + CbrBC2f	-	-	-	-	-	-	60.71	53.36	56.80	-	-	-
	Abner + CbrBC2ymf	90.13	80.46	85.02	63.58	69.46	66.39	60.99	49.78	54.81	57.68	67.50	62.20
	Banner + CbrBC2	90.14	73.56	81.01	60.71	73.12	66.34	57.24	39.01	46.40	52.13	74.53	61.35
	Banner + CbrBC2y	89.45	74.71	81.42	-	-	-	-	-	-	-	-	-
	Banner + CbrBC2m	-	-	-	62.62	72.04	67.00	-	-	-	-	-	-
	Banner + CbrBC2f	-	-	-	-	-	-	61.98	53.36	57.35	-	-	-
	Banner + CbrBC2ymf	89.04	74.71	81.25	63.23	72.47	67.54	63.43	49.78	55.78	55.28	72.81	62.85
three taggers	**Abner + Banner + CbrBC2**	89.79	80.84	85.08	57.67	74.41	64.98	52.25	41.70	46.38	50.37	75.31	60.36
	Abner + Banner + CbrBC2y	89.54	81.99	85.60	-	-	-	-	-	-	-	-	-
	Abner + Banner + CbrBC2m	-	-	-	59.11	73.98	65.71	-	-	-	-	-	-
	Abner + Banner + CbrBC2f	-	-	-	-	-	-	58.33	56.50	57.40	-	-	-
	Abner + Banner + CbrBC2ymf	89.54	81.99	85.60	59.52	74.62	66.22	59.20	53.36	56.13	52.72	74.22	61.65

Table 5.6: Comparison of the ensemble of taggers for the exact matching.
Comparative results are shown for the gene/protein normalization task using the exact matching, disambiguation by single decision and cosine similarity and according to the combination of taggers that were used for the extraction of gene/protein mentions. The results are separated according to the number of taggers used (1, 2 or 3). The best F-Measure is shown in gray.

In the case of the fly, the results in Table 5.6 show that the top best gene normalization F-Measures (shown in bold) are obtained always when using the CBR-Tagger trained with specific fly documents, in addition of the default training dataset for the BioCreative 2 Gene Mention task. An important increase in the recall and F-Measure is verified when using the CbrBC2f configuration, no matter the number of taggers considered (1, 2 or 3).

The mouse is the only organism whose performance is not always improved when using a tagger trained with organism's specific documents. By analyzing some of the mentions that are extracted by the CBR-Tagger when trained only with the BC2 dataset (CbrBC2) but not by the CBR-Tagger when trained also with the mouse's specific training documents (CbrBC2m), we have noticed that the source of the problem might be due to the case of the letters. In most of the mistakes, CbrBC2 was able to extract it because all of its cases in the known bases are gene mention, or at least most of them. However, in CbrBC2m's known cases, some extra non gene mention cases have been added from the organism's specific training documents.

By analyzing some of the documents that are part of the mouse's training dataset, we have noticed that these non gene mention examples have different cases from the positive ones. But, as the CBR-Tagger also considers a case insensitive approach (when the case sensitive search fails) the system ends up not classifying the mentions properly. This is the case of the "Apob" and "Gnrh" synonyms that have been placed together in the same case of the non-mentions "apoB" and "GnRH", and as consequence of this, they were not able to be recognized by the CbrBC2m, because negative examples of this token have appeared with much more frequency in the mouse specific documents than the positive ones.

Different mixes of taggers perform best for each organism under consideration here. In order to be able to claim that our methodology that is able to perform reasonably for any organism, a single configuration of the taggers must be adopted the one to be defined as the default one. The mix of ABNER, BANNER and CbrBC2ymf can be considered as the best alternative and it is the mix of taggers used for the results in Table 5.5.

The results show that the exact matching performs reasonably well to all organisms. The performance of our system may be well below the best results achieved in the past BioCreative competitions, in which the participating systems have made use of specific knowledge for each of the organism considered, which is not always available to the scientific community. Our intention was to construct a system that could perform reasonably well to any organism by providing the minimum organism-specific information as possible. Our experimental results prove that our approach is suitable to cope with these complex tasks in a very satisfactory manner.

An analysis of the errors for the mouse, fly and human organisms has shown that the mistakes are due to a variety of reasons, as for example, the tagger, the disambiguation

and the matching strategies. Table 5.7 shows details of the true positives, false positives and false negatives as well as if the mention was ambiguous or not. In the case of the false positives, the mistakes are usually due to mentions that were incorrectly extracted by the taggers but were able to be matched against one of the synonyms of the organism in consideration, due to our matching strategy using edited mention and synonyms.

These mentions resulted in false positive mistakes due to a wrong matching to other gene/protein or maybe because they are mentions from other organism or not really gene/protein mentions. For example, for the yeast, more than 80% of the false positive that matches to a unique gene/protein (no disambiguation performed) resulted in this class of mistakes. Some mistakes are also due to the disambiguation procedure. As the BioCreative competition allows only one identifier for a mention, instead of a ranking of candidates, sometimes the correct identifier is the second or the third option in the disambiguation procedure, but not the first one.

Analysis of results	**Unique Ambiguous**	**Yeast**		**Mouse**		**Fly**		**Human**	
		#	%	#	%	#	%	#	%
True Positive	**Unique**	501	98.0	349	82.7	157	53.4	503	42.3
	Ambiguous	10	0.02	73	17.3	137	46.6	150	12.6
False Positive	**Unique**	23	88.5	121	55.5	67	32.8	246	20.7
	Ambiguous	3	11.5	97	44.5	137	67.2	289	24.4
False Negative	**-**	101	-	122	-	134	-	131	-

Table 5.7: Error analysis for the exact matching.
The number of true positives, false positives and false negatives are shown for the evaluation using the BioCreative 1 (yeast, mouse, fly) and 2 (human) test dataset. The percentage of the true positives and false positives which made used of the disambiguation step is also shown.

The false negative mistakes are mostly due to differences between the mentions extracted from the text and the synonym in the dictionary, i.e., a matching problem. For example, the "Gastric alcohol dehydrogenase" (human Entrez Gene 131) could not be matched to the mention "recombinant human stomach alcohol dehydrogenase". In this case, "gastric" and "stomach" are clearly synonyms but are not easily learned by the system.

An example of matching that requires no great effort to be detected by an organism-specific system is the mention "hMAST205" to the synonym "MAST205" (human Entrez gene 23139) as a lower case "h" at the start of a upper case symbol usually stands for the human organism. As we attempt to develop a system the least dependent possible of manual organism-specific information, this case may not be present in the dictionary of synonyms.

Some false negative mistakes were also due to the fact that the mentions were not able to be extracted from the text, even when using a mix of three taggers. Some examples of these mistakes are "VIP-36", "Phosphatidylinositol-specific (PI-specific)

phospholipase-C" and "UbcH5" and also some mentions composed by a range, e.g. "LERK-2 and -5", in which only part of the mention was able to be extracted. The experiment carried out for the human with the gold standard mention confirms the importance of the tagger step in the performance of the normalization step.

5.3 Approximated matching based on trie and global alignment

In this section we describe the second method we propose for gene/protein normalization. It starts using an exact matching between the mentions with the synonyms of the dictionary, the faster matching strategy possible. For this exact matching, we use the original dictionary (cf. 5.1) which was previously only converted to lower case. In case the exact matching fails, we perform an approximated matching based on a global alignment of the mentions extracted from the text and the synonyms of the dictionary (Neves, Chagoyen et al. 2008). The dictionary of synonyms considered for both matching strategies is exactly the one described in section 5.1. No operations such as exclusion of punctuations, conversion of number and Greek letters are performed on the synonyms in order to have it as similar as possible to the original dictionary provided by BioCreative (cf. 5.1). The only editing operation was to convert the synonyms to lowercase.

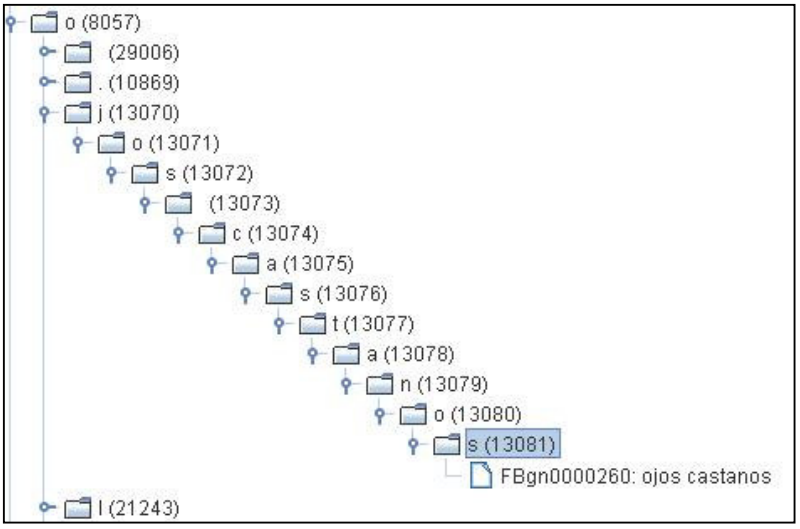


Figure 5.2: Example of the dictionary of synonyms represented as a trie.
The example is presented for the synonym "ojos castanos" in which each character of the synonym is represented as a node in the tree.

Usually, an approximated matching requires each mention to be compared to each synonym in the dictionary, which may be time-consuming. In order to overcome this problem and in order to improve the performance of the system, the synonyms have been built as a trie structure (a retrieval tree) (Shang and Merrettal 1996). In a trie, each token is represented by nodes of a single character according to a tree structure. Tokens with the same prefix are located in common branches of the tree. Figure 5.2 shows an

example of the branches for the fly synonym "ojos castanos" (FlyBase identifier FBg0000260).

The advantage of using a trie is that there is no need of performing repeated alignment operations when comparing a mention with synonyms that share the same prefix. Also, a search through a certain branch may be aborted if the minimum cost of the alignment at this branch is higher than a predefined threshold. The result of this strategy is a reduction in the processing time without sacrificing the quality of the comparison between the mention and the synonyms.

Our approximated matching performs a global alignment based on predefined costs, as suggested in (Tsuruoka and Tsujii 2003) for the gene recognition problem. The global alignment is carried out using the edit distance between the two strings (a mention and a synonym). This is performed using a dynamic programming algorithm and it may be better defined as the minimum number of operations (exclusion, insertions and substitutions) needed to be carried out on character level in order to transform one string into the other, as showed in Figure 5.3.

The initial costs for character substitution, inclusion and deletion were the ones proposed in the mentioned paper (Tsuruoka and Tsujii 2003). These costs were further adapted according to experiments carried out with the BioCreative yeast, mouse, fly and human datasets (cf. B.2 and B.3) during the development of the method. The final costs for each operation are presented in Table 5.8.

		G	R	-	2
	0	1	2	3	4
E	1	1	2	3	4
G	2	1	2	3	4
R	3	2	1	2	3
-	4	3	2	1	2
1	5	4	3	2	2

Figure 5.3: Dynamic programming for the comparison of two strings.
Example extracted from (Tsuruoka and Tsujii 2003) which illustrate the edit distance between two strings. Here the costs for is 1 (one) for the operations of exclusion, inclusion and substitution. The global alignment between the two strings is given by the right most and lower-most cell of the table which contains the value of "2".

When a mention is compared against a synonym, the lower-most and right-most cell of the dynamic programming matrix is the final result of the matching. To this cost we add 0.4, a constant value proposed by (Tsuruoka and Tsujii 2003), and normalize it by the length of the synonym under consideration. If this normalized cost is lower than a

parameterized threshold (defined as 3.0), the matching is considered a valid one. These parameters were defined while carrying out experiments with the BioCreative development datasets.

Characters	Inclusion Deletion	Characters	Substitution
Numeral	50	**Numeral by numeral**	50
Punctuation/Space	1	**Numeral by letter**	100
Letter "s" (plurals)	10	**Punctuation/Space**	1
First letter "h" (human)	10	**Letter by letter**	50
Last letter "p" or "c" (yeast)	10	**else**	50
else	50		

Table 5.8: Costs for the edit distance between a mention and a synonym.
The costs are shown for operations of inclusion, exclusion and substitution for the gene/protein normalization task. Some of these costs are specific for a certain organism.

The comparison is performed according to the alphabetical order of the branches of the trie. The strategy consists of trying to reach as deep as possible inside a branch. The exploration of a determined branch is interrupted at any point of the trie if none of the values in the dynamic programming matrix is lower than the threshold value. That is, when there is no more possibility of finding a matching with the given mention for any synonyms which hangs from the branch under consideration.

The results obtained with the BioCreative datasets are summarized in Table 5.9. The table presents the precision, recall and F-Measure values and compares the last of them to the best results for each organism, the same ones presented in the previous section. The best results for yeast and fly were obtained using the BioCreative task 1B (Hirschman, Colosimo et al. 2005) and for mouse and human were obtained using GNAT (Hakenberg, Plake et al. 2008).

	Yeast	Mouse	Fly	Human
Precision	94.92	64,30	44.24	54.83
Recall	88.42	76,47	56.41	81.66
F-Measure	**91.55**	**69,86**	**49.59**	**65.61**
Best FM	**90.1**	**81.10**	**81.40**	**83.31**

Table 5.9: Results for the gene/protein normalization using trie and global alignment.
Results are shown for recall, precision and f-measure for the yeast, mouse, fly and human. The best line of the table shows the best f-measure obtained so far, during the BioCreative Task 1B for the yeast and fly and by the GNAT system for the mouse and human.

The results reported in Table 5.9 are promising. The yeast F-Measure is almost as high as the best BioCreative results for some organism. The human recall is also high enough while the remaining values wouls still need to be improved. The fly recall is particularly low due to the fact that the taggers were not able to extract all the correct mentions from and documents, and consequently, these mentions could not have been normalized by the system.

5.4 Approximated matching based on machine learning

This last matching strategy we propose was modelled as a binary classifier based on three machine learning algorithms: support vector machines (Joachims 1998), random forests and logistic regression. It is carried out only if the exact matching fails. Here, we consider the exact matching as the perfect matching between the mention and the synonyms (cf. 5.1). That is, no editing operating is applied to any of them, as carried out in sections 5.2 and 5.3.

We use the Weka tool (Witten and Frank 2005) for training and testing the three machine learning algorithms. Weka implements many of the machine learning algorithms for data mining tasks which are available through a Java API or a system graphical interface. The training procedure consists in associating each example in the dataset with a set of features and its respective category. It also contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. In the testing step, examples are presented to the system and they are classified according to what has been learned during the training step.

For the training step, our features represent the comparison between a pair of synonyms and the category, i.e., whether it is a match or not. A machine learning model is constructed based on these examples. On the other hand, in the testing step, the features represent a comparison between a given mention and the sysnonyms of the dictionary. The model previously trained would classify each example as being a match or not.

The training of the machine learning matching is a three-step procedure in which the data produced in each phase are retained for further use. In order to construct a training set for the machine learning algorithm, we use the methodology proposed in (Tsuruoka, McNaught et al. 2007). The attributes of the training examples are obtained by a comparison of two synonyms in the dictionary according to some predefined features. For each organism, the comparison of two synonyms of a given gene will constitute the positive examples while the comparison of two synonyms of different genes from the same organism will result in the negative examples. The machine learning algorithm would then learn from these examples.

The steps for the construction of the training data include the extraction of features from the synonyms, followed by the selection of the pairs of synonyms to be compared. In the first step, the features that represent a synonym were extracted for all the synonyms of the dictionaries. This is carried out just once, during the development of the method, and the features are the following:

- prefix composed of the first three letters of the synonym (e.g., "hys" from "hyst2477");
- suffix composed of the last three letters of the synonym (e.g., "ase" from "adenosine deaminase").
- number that is part of the synonym (e.g., "2" from "cdh2");
- Greek letter that is part of the synonym (e.g., "gamma" from "pkb gamma");
- bigram and trigrams (cf. 3.1.3) of the synonym (e.g. "pna" and "nat" from "pnat");
- shape of the synonym (e.g. "a1a1" for "d9mit94").

The shape is a string value which represents the orginal synonym using only two characters: "a" and "1". Letters are substituted to "a" and numbers to "1", no repetition allowed. Any other characters, such as symbols and punctuations, are kept exactly as they appear in the sequence.

The second step is the selection of a set of pairs of synonyms to be compared, which will compose the positive and negative examples used for training the machine learning algorithm. This is a time-consuming step and the data obtained are stored for further use. The pairs of synonyms are selected in order to have a balanced training dataset, i.e., the same number of positive and negative examples, and so avoid the overabundance of one of the classes. In addition, synonyms were selected in such a way that a pair should share some similarity.

For each synonym in the organism's dictionary, the system selects positive (that belongs to the same gene/protein) and negative (that belong to a different gene/protein) pairs of synonym that share at least a certain percentage of bigrams or trigrams (cf. 3.1.3). The bigram/trigram similarity is given by Equation 5.1.

$$GS = \frac{2 \cdot \left| grams_1 \cap grams_2 \right|}{\left| grams_1 \right| + \left| grams_2 \right|}$$ **Equation 5.1**

We have set the maximum number of pairs of synonyms per organism to 30,000, half of them positive and the other half negative. Experiments were accomplished in order to decide the bigram/trigram similarity, ranging from 0.6 to 0.9, as presented in Table 5.10 and Table 5.11.

The features that compose the training examples used by the machine learning algorithm are not the ones listed above, which represent the synonyms themselves, but the features which represent a comparison between to synonyms, a pair of synonyms. Therefore, they are obtained by the comparison of the synonyms features shown above. This is the third step of the methodology. These features are the following:

- indicative of equal prefixes (0 or 1);
- indicative of equal suffixes (0 or 1);
- indicative of equal number (0 or 1);
- indicative of equal Greek letter (0 or 1);
- bigram or trigram similarity, given by Equation 5.1 and described below;
- string similarity, obtained by one of the available string distance methods;
- shape difference, as defined in Equation 5.2 below.

The bigram/trigram similarity is defined by the quotient between the double of the number of bigrams (or trigrams) in common in both synonyms and the sum of the number of bigrams (or trigrams) of each synonym, as presented in Equation 5.1 (Dice similarity) (Tsuruoka, McNaught et al. 2007). In this equation, "grams1" and "grams2" are the arrays composed by all the distinct bigrams or trigrams of the synonym, no repetition allowed.

The string similarity is a floating point value which represents the degree of similarity between the synonyms under consideration. There are many metrics for string distance comparison and some of them have even been used as the only matching strategy itself (Crim, McDonald et al. 2005; Neves, Chagoyen et al. 2008) (cf. 5.3). Here, string similarity will be used as one of the features of the training examples. We evaluated several string similarity metrics, such as Levenstein, Jaro-Winkler, Monge-Elkan, Smith-Waterman and SoftTFIDF (Cohen, Ravikumar et al. 2003), as presented in Table 5.10 and Table 5.11.

SecondString[19] (Cohen, Ravikumar et al. 2003) is an open-source Java-based package of approximate string-matching techniques. It includes a variety of string distance techniques. Some of the methods are based on edit distance, which assigns costs for the operation of insertion, deletion and substitution, or on the token that compose each string under comparison. The following methods were the ones which we have considered here. More details on each of them are presented below.

- Levenstein distance: it is based on the edit distance and a unit cost is considered for the operations of insertion, deletion and substitution.

- Smith-Waterman: this is a dynamic programming algorithm which is well-known for performing sequence alignment between two proteins or nucleotide sequences. Instead of looking at the total sequence, it compares segments of all possible lengths and optimizes the similarity measure.

- Monge-Elkan: this is a variant of the Smith-Waterman algorithm which uses some particular cost parameters scaled to the interval [0,1]. Also, the algorithm assigns lower cost to a sequence of insertion and deletions.

[19] http://secondstring.sourceforge.net/

- Jaro-Winkler: it is based on the number and order of the common characters between two strings and it is intended for short strings.

- Jaccard similarity: this is a token-based similarity which, given two strings (words sets), calculates the relationship between the tokens that are shared by both strings and the total number of tokens in both strings.

- TF-IDF: it is based on the TF (term frequency), the frequency of a word in a corpus, and on the IDF (inverse document frequency), the inverse of the fraction of elements in the corpus that contain a certain word.

In addition to the features proposed by (Tsuruoka, McNaught et al. 2007), we have added a new feature, the shape difference. It is a floating point value calculated by the number of differences between the shapes of each synonym in the comparison. The shape difference is presented in Equation 5.2 in which "diff" is the number of differences between both shapes, i.e., the number of characters in a sequence of symbols that does not match the other shape. The value of this difference is doubled and divided by the sum of the lengths of both shapes.

$$SD = \frac{2 * diff}{|sh_1| + |sh_2|}$$ **Equation 5.2**

For the evaluation of the methodology, mentions are extracted using appropriate tagger(s). The system then repeats the three-step procedure for each mention: the features of the mentions are extracted (synonym-features); the system selects the candidate synonyms according to a certain percentage of bigram/trigram similarity between the synonyms and the given mention; and finally, the features of the selected pairs (pair-features) are extracted. Therefore, only those synonyms which share some degree of bigram or trigram similarity are chosen, and their features are compared to the mention features. The features that represent this comparison are sent to the previously trained machine learning algorithm that will decide which pairs match, i.e., which pairs result in a positive classification. If a pair of mention-synonyms is classified as positive, the identifier of the respective synonym is set as the gene/protein identifier of the given mention and the normalization task is over.

There are many parameters which may be configured when using machine learning algorithms. Weka provides methods for setting most of them. In our experiments, we have decided for the default parameters for each of the Java classes that implement the three algorithms under consideration here. If more than one pair (mention-synonym) results as positive for the same mention, a disambiguation step is followed (cf. 5.5). This procedure is repeated to each mention extracted from the tagger(s).

The first step that should be taken into account is the selection of features to be used in the training of the machine learning algorithms. Due to the high cost of training and testing the different combination of features for the four organisms, experiments have been accomplished only for the yeast and fly, organisms with a simple and complex nomenclature for its synonyms, respectively. Table 5.10 and Table 5.11 show the results of the experiments performed for the yeast and fly organism, respectively. Results are compared to the exact matching methodlogy describe in section 5.4.

Exact matching	P	R	FM
	89.54	81.99	85.60

YEAST		**SVM - 0.6**			**SVM - 0.7**			**SVM - 0.8**			**SVM - 0.9**		
Results		P	R	FM	P	R	FM	P	R	FM	P	R	FM
F1	**Levenstein**	49.02	86.21	62.50	58.42	87.74	70.14	69.66	86.21	77.05	76.43	81.99	79.11
	Jaro-Winkler	46.06	85.06	59.76	58.42	87.74	70.14	69.44	86.21	76.92	76.43	81.99	79.11
	Smith-Waterman	49.89	86.21	63.20	58.27	87.74	70.03	69.66	86.21	77.05	76.98	81.99	79.41
	Monge-Elkan	48.91	85.82	62.31	58.42	87.74	70.14	69.66	86.21	77.05	76.98	81.99	79.41
	Soft-TFIDF	56.46	85.44	67.99	58.42	87.74	70.14	69.66	86.21	77.05	76.43	81.99	79.41
F2	**Levenstein**	48.08	86.21	61.73	61.54	85.82	71.68	72.58	86.21	78.81	76.70	81.99	79.26
	Jaro-Winkler	48.08	86.21	61.73	61.54	85.82	71.68	76.53	86.21	81.08	**84.92**	81.99	**83.43**
	Smith-Waterman	49.89	**88.12**	63.71	77.40	86.59	81.74	71.43	86.21	78.13	76.70	81.99	79.26
	Monge-Elkan	48.50	86.59	62.17	61.54	85.82	71.68	73.53	86.21	79.37	76.98	81.99	79.41
	Soft-TFIDF	56.17	85.44	67.78	74.83	86.59	80.28	75.25	86.21	80.36	76.70	81.99	79.26

Table 5.10: Detailed results for the yeast according to the machine learning features.
Results are shown for the exact matchingand for the support vector machine (SVM) strategy according to the following parameters: features F1 and F2, percentage of similarity (from 0.6 to 0.9) and five string matching techniques.

For the results presented in Table 5.10 and Table 5.11, the mentions have been extracted using the mix of Abner (cf. C.5), Banner (cf. C.6) and CbrBC2ymf (cf. 4.6) and the default disambiguation methodology (single disambiguation based on cosine similarity and the number of common words) (cf. 5.5). The algorithm used to train the models was Support Vector Machines implemented in Weka tool (SMO class) and the selection of the pairs of features was based on the bigram and trigram similarity. These experiments were intended to decide the following parameters of the machine learning configuration:

- string similarity method: Levenstein, Monge-Elkan, Jaro-Winkler, Smith-Waterman or SoftTFIDF, all of them implemented in SecondString Java library (Cohen, Ravikumar et al. 2003);
- percentage of similarity, that ranged from 0.6 to 0.9;

- and set of features:
  - F1 (if all of the features discussed in here): trigram similarity, bigram similarity, 3-letters prefix, 3-letter suffix, number, Greek letter, shape and string similarity
  - F2 (just the set of the more meaningful ones): trigram similarity, bigram similarity, number and string similarity).

Exact matching	**P**	**R**	**FM**
	58.50	52.47	55.32

FLY		**SVM - 0.6**			**SVM - 0.7**			**SVM - 0.8**			**SVM - 0.9**		
Results		P	R	FM	P	R	FM	P	R	FM	P	R	FM
F1	**Levenstein**	19.91	57.40	29.56	22.62	56.50	32.35	27.23	57.40	36.94	34.92	56.05	43.03
	Jaro-Winkler	22.51	58.74	32.55	24.11	57.85	34.04	28.13	57.40	37.76	50.00	56.95	53.25
	Smith-Waterman	21.80	**59.64**	31.93	24.11	57.85	34.04	28.13	57.40	37.76	52.52	56.05	54.23
	Monge-Elkan	29.31	58.74	39.10	32.58	58.30	41.80	36.59	58.74	45.09	**52.97**	56.05	**54.47**
	Soft-TFIDF	26.77	59.19	36.87	34.03	58.74	43.09	39.09	57.85	46.65	52.74	56.05	54.35
F2	**Levenstein**	21.80	**59.64**	31.93	24.11	57.85	34.04	28.13	57.40	37.76	35.41	56.05	43.40
	Jaro-Winkler	21.80	**59.64**	31.93	24.11	57.85	34.04	28.13	57.40	37.76	36.81	56.95	44.72
	Smith-Waterman	21.80	**59.64**	31.93	24.11	57.85	34.04	28.13	57.40	37.76	52.52	56.05	54.23
	Monge-Elkan	25.90	58.30	35.86	24.11	57.85	34.04	28.13	57.40	37.76	52.08	56.05	54.00
	Soft-TFIDF	26.65	57.85	36.49	31.76	57.40	40.89	38.96	56.95	46.27	50.81	56.05	53.30

Table 5.11: Detailed results for the fly according to the machine learning features.
Results are shown for the exact matchingand for the support vector machine (SVM) strategy according to the following parameters: features F1 and F2, percentage of similarity (from 0.6 to 0.9) and five string matching techniques.

The selection of the best features was performed with Weka's feature selection functionalities. The features selection methods we have tried in Weka in order to decide the best features were the one based on chi-squared statistics and on the gain ratio to measure the attributes individually. These methods are implemented in Weka 3.4.11 by the names of "ChiSquaredAttributeEval" and "GainRatioAttributeEval", respectively. The scores of the features for each of these feature selection methods are presented in Table 5.12 and Table 5.13. The more meaningful features are the one with higher scores (or gain) and these were the ones that compose the F2 set of features.

There is not a clear pattern of the influence on the results of a string distance method and the set of features (cf. Table 5.10 and Table 5.11). For the yeast, the best results were when using Jaro-Winkler and just the more significant features (F2) while the fly performs better with Monge-Elkan and Smith-Waterman and all the features considered here (F1). But both organisms coincide that the higher the percentage of similarity, the higher the F-measure. The chosen configuration that performs reasonably well to both organisms was decided to be the Smith-Waterman and the selected set of features (F2), along with the percentage of similarity set to 0.9. Smith-Waterman is not the string

similarity that performs best to any of the organisms but is the one that along with Soft-TDIDF outputs a reasonable F-Measure for both the yeast and fly.

Features	Mouse		Fly		Human	
	score	order	score	order	score	order
Trigram similarity	**17037.3204**	**1**	5552.5994	2	2852.9692	4
Bigram similarity	11576.0703	5	**6287.0158**	**1**	3327.318	3
3-letters prefix	15005.5936	2	2275.9809	6	3909.222	2
3-letters suffix	0	8	429.5092	7	850.3518	6
Number	12951.5364	3	4031.3882	4	**5017.1205**	**1**
Greek letter	24.6761	7	0	8	152.9694	8
Shape	328.4629	6	3237.3631	5	780.3099	7
String similarity	11877.5605	4	4174.1497	3	1115.0981	5

Table 5.12: Feature selection using the ChiSquaredAttributeEval method.
Score for each feature using the "ChiSquaredAttributeEval" class of Weka.

In summary, many experiments have been accomplished for the four organisms (yeast, mouse, fly and human) in order to compare the influence of the various parameters here considered: mix of taggers, machine learning algorithm, string distance metrics for the string similarity feature and feature selection. It has been noticed that although different configurations of taggers would be necessary for achieving the best results for an specific organism, good enough results may be obtained by using the Abner+Banner+CbrBC2ymf as the mix of taggers and the disambiguation based on a single solution and the cosine similarity combined with the number of common words between the text and the gene-documents (cf. 5.5). The results show that the exact matching (cf. 5.2) performs better to all organisms, although some gain in the recall is sometimes observed when using a machine learning matching, especially for the mouse and human.

Features	Mouse		Fly		Human	
	score	order	score	order	score	order
Trigram similarity	0.1138	3	0.0812	3	0.01744	4
Bigram similarity	0.0718	4	0.078	4	0.01695	5
3-letters prefix	**0.46415**	**1**	0.083	2	0.10075	2
3-letters suffix	0	8	0.0188	7	0.03508	3
Number	0.44821	2	**0.1769**	**1**	**0.12586**	**1**
Greek letter	0.00452	7	0	8	0.00804	8
Shape	0.04813	6	0.0662	5	0.01571	6
String similarity	0.06561	5	0.0401	6	0.00975	7

Table 5.13: Feature selection using the GainRatioAttributeEval method.
Score for each feature using the "GainRatioAttributeEval" class of Weka.

Organism	Best results (BioCreative and GNAT)			Moara results Machine learning matching		
	Recall	Precision	F-Measure	Recall	Precision	F-Measure
Yeast	89.4	95.0	92.1	84.34	81.67	82.99
Mouse	91.6	72.6	81.0	79.60	32.90	46.56
Fly	80.0	83.1	81.5	69.00	55.22	61.35
Human	90.1	81.1	85.4	85.99	29.13	43.52

Table 5.14: Results for the approximated matching based on machine learning.
Best results by organism for the gene/protein normalization task evaluated with the test corpora of the BioCreative 1 task 1B (yeast, mouse and fly) and BioCreative 2 Gene Normalization task (human). The results were produced using a mix of Abner, Banner and CBR-Tagger (CbrBC2ymf), flexible matching, and single disambiguation by cosine similarity multiplied by the number of common words. The machine learning configuration uses Support Vector Machines, the F2 set of features (trigram similarity, bigram similarity, number and string similarity), pairs of synonyms selected by 0.9 trigram and bigram similarity and Smith-Waterman for the string similarity feature. The best results for each organism in both competitions are shown.

In the case of the machine learning matching, experimental decision was made for selecting the pairs of synonyms based on both the bigram and trigram similarity (percentage of similarity of 90%), the consideration of the best selected features only (trigram similarity, bigram similarity, number and string similarity) and Smith-Waterman as the string similarity feature. Table 5.14 presents the results for the organism under consideration here, for the machine learning matching, along with the best results. The best results for yeast and fly were obtained using the BioCreative task 1B (Hirschman, Colosimo et al. 2005) and for mouse and human were obtained using GNAT (Hakenberg, Plake et al. 2008).

Analysis of results	Exact / Machine Learning	Unique / Ambiguous	Yeast		Mouse		Fly		Human	
			#	%	#	%	#	%	#	%
True Positive	**Exact**	**Unique**	496	96.1	318	73.4	153	51.9	436	64.7
		Ambiguous	10	1.9	62	14.4	131	44.4	137	20.3
	Machine Learning	**Unique**	5	1.0	19	4.4	6	2.0	22	3.3
		Ambiguous	5	1.0	34	7.8	5	1.7	79	11.7
False Positive	**Exact**	**Unique**	23	19.8	117	13.3	66	27.5	237	14.4
		Ambiguous	3	2.6	101	11.5	136	56.6	293	17.6
	Machine Learning	**Unique**	24	20.7	168	19.0	22	9.2	231	13.9
		Ambiguous	66	56.9	496	56.2	16	6.7	899	54.1
False Negative	**-**	**-**	96	-	111	-	133	-	110	-

Table 5.15: Error analysis for the approximated matching based on machine learning
The table shows the number of mistakes (#) and the percentage (%) regarding the total of true positives, false positives and false negatives. Results are also shown regarding the type of matching (exact or machine learning) and whether matching only one entry (unique) or more than one in the dictionary.

Although machine learning matching often produces poorer results than exact matching, it is a useful alternative when working with new organisms where the user has no indication of the performance of exact matching. In addition, machine learning produces better recall performance than exact matching, although it is not as precise. In cases where higher recall is needed, machine learning is the best alternative to use.

An error analysis for the gene/protein normalization task is showed in Table 5.15 as well as the statistics of the contribution of the disambiguation and the approximated matching procedures in the results. It can be noticed, that due to the simplicity of the yeast nomenclature, most of the true positive results (96.1%) are matched against only one synonym and are carried out by the exact matching. On the other hand, the fly has the more complex nomenclature and about half of the true positives are matched to a unique synonym and half of them are ambiguous.

5.5 Disambiguation of the Identifiers

When more than one identifier is obtained for a mention, a disambiguation procedure is used to decide which is more likely to be correct. The selection decision is performed by comparing the similarity between the abstract of the article and a document representative of each of the genes/proteins (gene-document). The gene-document is constructed by compiling information extracted from several databases, such as SGD[20] (Cherry, Adler et al. 1998) (cf. A.5) for yeast, MGI[21] (Eppig, Bult et al. 2005) (cf. A.6) for mouse, FlyBase[22] (Gelbart, Crosby et al. 1997) (cf. A.7) for the fly and Entrez Gene[23] (Maglott, Ostell et al. 2007) (cf. A.3) for humans. The fields collected for the construction of the gene-documents were symbols, aliases, descriptions, summaries, products, phenotypes, relationships, interactions, Gene Ontology[24] (Ashburner, Ball et al. 2000) (cf. A.4) terms related to the gene and their names, definition and synonyms.

The text contained in theses fields is tokenized and the resulting tokens are grouped together into a bag of words. A vector space model is constructed to each document and it is composed by all its tokens, except cardinal and ordinal numbers, pre-defined unit measures (such as “10-kb”) and tokens that match with a stopwords list (cf. Appendix E.1). The resulting tokens are reduced to their stem with a freely available Java implementation of the Porter stemmer (cf. C.2) and weighted in the document according to the TF-IDF measure (Salton and Buckley 1988). This procedure is performed to the each gene-document of the candidates as well as for the article in consideration.

Three disambiguation methodologies can be selected. The first considers the cosine similarity (Shatkay and Feldman 2003) between the article and the gene-documents,

[20] http://www.yeastgenome.org/
[21] http://www.informatics.jax.org/
[22] http://flybase.org/
[23] http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene
[24] http://www.geneontology.org/

while the second takes into account the number of common tokens between the two texts. In the first case, the gene-document with the highest cosine similarity is chosen as the correct identifier for the mention. In the second case, the gene-document with highest number of common tokens is chosen as the best solution. The third methodology, based the decisions on both the higher product of the cosine similarity and the number of common tokens, is the default option.

For the decision of the best disambiguation strategy, some experiments have been carried out and they have also taken in consideration the selection of only the best candidate (single disambiguation) or the top scoring ones according to a given threshold (multiple disambiguation). The latter threshold is automatically calculated for each ambiguous mention and is given by 50% of the value of the highest score candidate. It means that the number of the chosen candidates may be different in each situation; it is not a determined value arbitrarily defined. For example, a mention was matched to four candidates with scores of 0.9, 0.7, 0.5 and 0.4. Using single disambiguation, the only answer is the candidate with best score, 0.9. Using multiple disambiguation, the threshold is automatically calculated as 50% of the highest score, therefore 0.45. The candidates with scores 0.9, 0.7 and 0.5 would be returned by the system as their scores are higher than the threshold.

Type Disambig.	**S/ M**	**Yeast**			**Mouse**			**Fly**			**Human**		
		P	**R**	**FM**	**P**	**R**	**FM**	**P**	**R**	**FM**	**P**	**R**	**FM**
None	**-**	**90.60**	81.23	85.66	**65.45**	61.94	63.65	**63.55**	30.49	41.21	**65.32**	62.97	**64.12**
Cosine	**S**	89.96	**82.38**	**86.00**	57.72	73.12	64.52	52.22	47.53	49.77	50.11	73.12	59.47
	M	89.21	**82.38**	85.66	46.75	77.42	58.30	18.22	57.85	27.71	30.10	78.59	43.53
Num words	**S**	89.54	81.99	85.60	59.04	74.41	65.84	55.45	50.22	52.71	50.42	74.53	60.15
	M	88.48	**82.38**	85.32	46.43	76.99	57.93	31.53	60.09	41.36	25.31	**79.06**	38.35
Cosine + Num words	**S**	89.54	81.99	85.60	59.72	74.62	**66.35**	58.50	52.47	**55.32**	52.04	75.62	61.66
	M	89.58	**82.38**	85.83	52.24	**77.63**	62.46	38.97	**60.99**	47.55	33.18	78.12	46.58

Table 5.16: Results for the gene/protein normalization according to the disambiguation. Comparison of results for the gene/protein normalization task when using flexible matching, mentions extracted using the combination of Abner, Banner and CBR-Tagger (CbrBC2ymf) and according to the disambiguation methodologies. A comparison was done using no disambiguation when ambiguous identifiers are not considered (None), and using single (S) or multiple (M) disambiguations. "P" stands for precision, "R" for recall and "FM" for f-measure. The highest value of precision, recall and f-measure is highlighted for each organism.

Experiments were performed in order to compare the disambiguation methodologies proposed here. Table 5.16 presents the results when using a exact matching (cf. 5.2) and extracting the mentions with a mix of ABNER (cf. C.5), BANNER (cf. C.6) and CbrBC2ymf (cf. 4.6) taggers. The results presented here where evaluated with the BioCreative Task 1B (Hirschman, Colosimo et al. 2005) and BioCreative 2 Gene Normalization Task (Morgan, Lu et al. 2008).

When not using the disambiguation step, the system does no take into account the ambiguous mentions, the ones that match with more than one identifier in the dictionary of synonyms. When more than one candidate exists, results may experience a decrease in the precision as each mention should be associated to only one identifier, according to the BioCreative gold standard annotations.

The above results clearly show the disambiguation step always increases the recall in more than 10 points, except for the yeast in which ambiguity is not a concern. However, in spite of the high improvement in the recall, some decrease in the precision is also verified, resulting in only a slight increase in the F-Measure for some organisms, as it is the case of the mouse, or no improvement at all, as it is the case of the human.

The use cosine similarity combined with the number of common words is the methodology that performs reasonably well for all the organisms here considered. The consideration of the multiple decision increase substantially the recall for all the organisms (except for the yeast), but the decrease in the precision is also very high, resulting in a poor F-Measure. The default configuration for the system is thus the single disambiguation using the cosine similarity combined with the number of common words between the text and the gene-documents.

5.6 Summary of the chapter

In this chapter we have described the proposed methodologies for the normalization of named entities, and specifically for the gene and protein mentions. The normalization is usually based on a dictionary of synonyms, which was introduced in section 5.1.

Three methodologies are proposed for the task. The first based on an exact matching (cf. 5.2), and two approximated matching based on a global alignment of the mention and the synonyms (cf. 5.3) and one based on machine learning (cf. 5.4).

On section 5.2 the exact matching is described which uses some pre-processing steps in order to make both the synonyms of the dictionary and the mention more flexible. The mentions and synonyms are therefore gradually cleaned according, for instance, to stopwords or biomedical terms.

The second methodology is presented in section 5.3 and it is based on an approximated matching between the mention and the synonyms, which is based on a global alignment between the both of them. We use a trie structure to represent the dictionary of synonyms.

The third methodology is based on machine learning methods (cf. 5.4) in which some positive and negative examples of synonym-mention are generated and for training a machine learning algorithm, which will be further be used to classify the unknown pair of mention-synonym as positive (matching) or negative (no matching). We have used three machine learning algorithms for this purpose, namely: Support Vector Machines, Random Forest and Logistic Regression.

Sometimes, more than one identifier is obtained for a mention, and the best candidate(s) should be selected as solution for the problem. The disambiguation of the identifiers has been discussed in section 5.5 and we propose a methodology based on the context in which the mention appears in the text and we compare the words that appear in the text with those which are representative of each gene or protein, which were obtained from a compilation of data from various freely available databases. The comparison is carried out using different methods, such as the cosine, the number of equal word and a mix of both.

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

In this work we present some novel methodologies for some text mining problem, more specifically for the named-entity extraction, extraction of biomedical relationships and normalization of named entities. For the first two tasks, a case based reasoning approach has been used and we start by discussing the performance of this method for the text mining tasks under analysis in this thesis.

In general, the methodologies proposed in this thesis have used little domain knowledge, with the exception of the disambiguation of gene and protein step (cf. 5.5), in which data extracted from genome databases specific for each organism was used. The decision for proposing methodologies which use little domain knowledge was mainly due to two reasons: the intentional decision of building a system as general as possible, without the need of biomedical experts, and the lack of the latters in our team, and therefore, the impossibility of acquiring such knowledge. The price paid to achieve this generality is, as expected, the sacrifice of performance, although our experimental results prove that our methods present satisfactory results in the tasks for which they were designed, while they could also be improved with domain knowledge.

Regarding the case based reasoning is cycle (cf. 4.1), our methodologies consider only the "retrieve" and "reuse" steps, when cases are retrieved from the base of cases and reused to give a solution for a new problem, respectively. The decision of not considering the "revise" step was due to the impossibility of getting a feedback from the users of the system and consequently, the incapability of saving the revised case for future use. Such revision could only be performed on a testing dataset, but such dataset should be used only for evaluation and its utilization for the revision and retained of a new case would not be accordance with the machine learning principles.

For the extraction of gene and proteins, a necessary step in many text mining procedures in biomedicine, we proposed a case based reasoning approach. Results show the suitability of this approach for the task. Although CBR-Tagger does not produce the best results when used alone, when combined with other taggers (such as ABNER or BANNER), our experiments showed that it improves the results for the gene and protein normalization task (cf. 5.2).

Although the results presented the for the gene mention extraction seem to indicate that training the system with organism-specific documents might result in a worse performance, the results presented for the normalization of genes/proteins for the fly (cf. 5.3) clearly shows that it is not the case. We consider that gene/protein recognition is a preceding step for the normalization problem, and the improvement of this one is the main goal of a gene/protein tagger. Also, the organism-specific documents used for training the tagger have only used the exact matching mentions with a dictionary of synonyms, and no extra knowledge related to the organisms was added to the system, which is an advantage in those cases where this domain specif information is not

available. Training the system with these documents may also help other organisms, as it is the case of the human.

As limitations of this approach, other features might have been used, such as some of the ones described in section 3.2.1, as well as a larger window of tokens. These limitations were usually due to time constraints as the recognition of gene and protein was the first methodology which has been developed as part of this thesis, initially for the BioCreative 2 Gene Mention task (Smith, Tanabe et al. 2008). The case based reasoning approach was then further improved, for the extraction of the biomedical events (cf. 4.4). In this improved implementation, retaining and retrieving of a case from the MySQL database is much more effective with the use of indexes and cache table which saves past results.

Regarding the normalization of genes and proteins (cf. Chapter 6), our methodologies may not reach the levels of other existing systems. However, as far as we know, for this task, no other freely available tool allows its integration and training with new organisms. This is a strong point in our work since it allows plenty of room for improvements. Once again, we could get a satisfactory F-Measure with no need of making adaptations in the algorithms to fit any particular organism. In addition to this, our system has been designed with very little dependency with custom dictionaries or annotated documents, which are generally not publicly available. When comparing our results with those reported in the two editions of BioCreative (Hirschman, Colosimo et al. 2005; Morgan, Lu et al. 2008), we have found that those which have achieved better F-Measure than ours have made use of an organism-specific procedure either for a curated dictionary or for the matching strategy.

Therefore, we can claim that our normalization system requires much less dependence on organism-specific knowledge as it uses only information that are freely available to the scientific community (online public databases), and no specific knowledge inferred from experts, as most of the other systems do. Most of the methods and tools which perform well for the gene normalization task usually use specific information for tuning the system and even to hardwired specific rules. Even if this approach produces good results for a specific organism, it cannot be extended to new organisms without a similar set of rules inferred from expert knowledge. Therefore, using or reproducing those existing methods with new organisms is very time consuming and sometimes even impossible. In our system, we use only publicly and general available information for every organism. It is true that we can not exclude some of the needed organism-specific information like the dictionary of synonyms or gene/protein annotations which are necessary for the matching and disambiguation tasks, respectively. However, this information can be obtained from public databases and no organism-specific tailoring is necessary to obtain satisfactory results.

In the case that a new organism is to be introduced in the system, a dictionary of synonyms and information related to its genes and proteins, such as description, phenotype, associated Gene Ontology, are the only necessary knowledge to be used. All this information is usually easily obtained from online organism-specific databases. The bottleneck here is the necessity of annotated documents for the evaluation of the results. That is the main reason that we could not extend our system with other organisms than the ones whose corpora are available in the BioCreative challenges. The availability of gene-relevant documents is currently a limitation of automatic text mining approaches, in particular those that require a collection of literature references relevant to genes and proteins under study and lack of manual annotations of associated bibliographic references.

We have analyzed the errors for the gene/protein normalization from the development dataset, as no analysis of the test set is accomplished so as to keep it as a blind test set. Some of the false negative mistakes were due to mentions that could not been extracted from the taggers, even when using a mix of three of them. Also, a high number of false negatives are due to a wrong disambiguation, with the consequent generation of many other false positives. Results provided with this approach look very promising and can certainly have more room for improvement, in particular with the disambiguation procedure. This procedure was not originally in the main focus of this study although results clearly indicate that more efforts should be devoted to those since global improvement heavily depends on its performance.

The application of our gene/protein normalization methodology to real data mining problems would require handling with more than one organism at the same time. The implementation of this functionality in the system is feasible as future work as it would only requires to perform the matching procedure with basic dictionaries of synonyms, specific to each organism in consideration, and disambiguate among them using a strategy similar to the one proposed in this paper. It is in this context where we hope our system serves as a good starting point which, besides producing good quality results, as shown throughout this paper, has a flexible structure to allow new ideas to be plugged in for improvement.

The final configuration of the system might be tailored by the user according to its need, in order to achieve a best precision, recall or F-Measure. We have implemented the methodologies proposed in this thesis in the Moara project (cf. F.1), a Java library freely available to be used by the scientific community. Moara includes classes that allow the user to test the CBR-Tagger the ML-Normalization described here, including the possibility of choosing the training documents used to train the tagger and the string similarity methods used as features in the machine learning matching strategy. Two version of the CBR-Tagger were integrated into U-Compare (cf. F.2) and we have plans to also integrate the ML-Normalization as there are few system for this task integrated into this framework.

We have also proposed a methodology for extracting biomedical relationships on the basis of case-based reasoning. We have evaluated our methodology with two domains, the extraction of disease and treatment relationships with the BioText corpus (cf. 4.5), and the extraction of biomedical events with the BioNLP Shared Task corpus (cf. 4.4). Our results show that the use of CBR is feasible for the relationship extraction problem and that the methodology proposed here returns satisfactory results for both corpora.

The analysis of the mistakes presented for the extraction of biomedical events confirms the complexity of the tasks, which includes the extraction of the trigger tokens. We conclude that our machine learning approach is satisfactory for this task but more experiments should be carried out and other features might need to be considered for both classifiers in order to enhance the performance of the system. In addition, the automatic analysis of errors is a hard task since no hint is given with the false positives and false negatives outputted by the evaluation system.

Regarding the limitations of our methodology for the relationship extraction, we assume for both domains that the named entities are given in the text. However, we think this is a fair assumption given the current performance of the named entity taggers. Also, and in order to reduce the processing time, we plan to make changes in the automatic generation of event candidates by adding some constraints, and, consequently, reducing the number of candidates contexts that need to be analyzed by the CBR classifier.

In the relationship extraction, the context is essential for the correct solution of the problem, especially when modalities, such as speculation and negation, are under consideration. One way of exploring more of the context of the sentence is by using deep parsing, which we did not exploited much in our methodology. Also, about half of the false positives and false negatives mistakes are due to trigger not being able to be extracted correctly and in this case, just as what was discussed for the extraction of gene and proteins, different features and a larger window of tokens might help in solving this problem. Some little domain knowledge, such as list of the most common event triggers might also help and it could be obtained automatically from the training data.

The extraction of the biomedical events is a good example of the levels of difficulties in the relationship extraction tasks. It contains types of event which can be easily extracted, such as the gene expression, whose nomenclature does not vary much. Also, the gene expression event is composed only of one argument, a theme, which is the protein. On the hand, the binding event may be composed of one, two or three themes, and it is, therefore, much harder to extract. And even harder are the regulatory events, which may have proteins or other events as arguments. Besides, the events may have modifiers such as speculation and negation. In our methodologies, we did not put much effort on extracting the modifiers. Additionally, the exploration of the context of the sentences and the use of deep parser may enhance the performance of the system for

this problem, as well as for resolving the co-references, which have not been taken into account here.

The methodology proposed for the extraction of biomedical events is not yet available in the Moara project, but it has also been integrated into the U-Compare framework (cf. F.2). It is part of joint server which allows a comparison of the results of some of the systems which participated in the BioNLP'09 Event Extraction shared task (Kim, Ohta et al. 2009). We also plan to evaluate our methodologies with some extra corpus. A new version of the Event Extraction shared task[25] has taken place during 2010/2011 and new documents are available, including some full texts. We also plan to try our methodologies for some binaries relationships, such as the protein-protein interaction (Tikk, Thomas et al. 2010) and the drug-drug interaction[26].

Regarding the implementation of Moara, we also plan to make a new version with more functions, specially the relationship extraction module. A better documentation of the library should be written as well as the API. Regarding the relationship extraction task, we plan to implement using a more flexible model, in order to allow it to be used for any type of relationship. Additionally, we could make available some predefined features ready to be used for the case-based reasoning approach.

In general, biomedical information extraction performs worse when compared to other domain, as for example newswire. Some authors (Zhou and He 2008) argue that one of the reason is that ontologies and terminologies are not well used or not used at all and that they are a prerequisite to obtain a good performance from the system. For the methodologies we propose in this thesis, this is a point to be explored in our future works as few or none ontologies were used, with the exception of the disambiguation step for the normalization of genes and proteins.

[25] http://sites.google.com/site/bionlpst/
[26] http://labda.inf.uc3m.es/DDIExtraction2011/

APPENDIX A: DATABASES AND TERMINOLOGIES

In this section will describe the available biomedical resources, such as online databases and ontologies, which have been used in the development of the any of the methodologies proposed in this work. They are the ones which have been mainly used for the construction of the dictionary of synonyms and the disambiguation step in the normalization of gene and protein task (cf. 3.4.1).

A.1 PubMed

PubMed[27] is a free database which contains more than 20 millions of citations for literature on life sciences and biomedical topics. It is considered the most important repository of scientific literature in the biomedical domain. PubMed allows searching the database by means of boolean operators or MESH (Medical Subject Headings) tags[28], a controlled vocabulary for indexing the PubMed repository. PubMed is widely used in the biomedical text mining domain as the main source of literature. Most of the documents used for the training and evaluation of our methodology are abstracts which have been extracted from PubMed.

A.2 BioThesaurus

The BioThesaurus[29] consists of a web-based system designed to map a comprehensive collection of protein and gene names to UniProt Knowledgebase (UniProtKB) protein entries. It provides dictionary of protein synonyms and some biomedical token-based dictionaries which can be freely downloaded form the web site. They include lists of biomedical, chemical, macromolecules and common English terms, among others. These terms have been used in this thesis for the automatic generation of synonym variations in the gene and protein normalization task (cf. 5.2).

A.3 NCBI Entrez Gene

Entrez Gene[30] (Maglott, Ostell et al. 2007) is the National Center for Biotechnology Information (NCBI) database for gene information and provides unique identifiers for genes for some of the model organisms. It usually includes the following information about the genes: nomenclature, map location, gene products and their attributes, markers, phenotypes, and links to citations, sequences, variation details, maps, expression, homologs, protein domains and external databases, among others. Data can be accessed through the web site or downloaded via FTP. The fields “Gene symbol”, “Gene description”, “Locus Tag”, “Preferred Names” and “Names” from this database have been used to construct the gene documents which are the basis of the

[27] http://www.ncbi.nlm.nih.gov/pubmed
[28] http://www.ncbi.nlm.nih.gov/mesh
[29] http://pir.georgetown.edu/pirwww/iprolink/biothesaurus.shtml
[30] http://www.ncbi.nlm.nih.gov/gene

disambiguation step (cf. 5.5) in the gene and protein normalization task. One of the databases which are part of the NCBI Entrez Gene is the NCBI Taxonomy[31] that provides information about the organisms.

A.4 Gene Ontology

The Gene Ontology[32] (GO) (Ashburner, Ball et al. 2000) is a collaborative project that aims to develop an ontology to describe gene products. The ontology covers three domains: cellular component, the parts of a cell or its extracellular environment; molecular function, the elemental activities of a gene product at the molecular level; and biological process, operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms. The data are usually accessed by AmiGO (the GO browser) or download via FTP. The fields "Name", "Synonyms", "Definition" and "Ontology" (biological process, molecular function or cellular component) from this database have been used to construct the gene documents which are the basis of the disambiguation step (cf. 5.5) in the gene and protein normalization task.

A.5 Saccharomyces Genome Database

The Saccharomyces Genome Database[33] (SGD) (Cherry, Adler et al. 1998) is a scientific database for the molecular biology and genetics of the yeast Saccharomyces cerevisiae, which is commonly known as baker's or budding yeast. The database includes genomic and biological information which are maintained by the SGD curators. The project also provides guidelines for the nomenclature of new genes for the yeast. The data is available though web browsing as well as via FTP. Some information from this database, such as the fields "Standard name", "Systematic name", "Alias", "Description", "Mutant Phenotype Free Text" and "Gene products", has been used to construct the gene documents which are the basis of the disambiguation step (cf. 5.5) in the gene and protein normalization task.

A.6 Mouse Genome Informatics

Similar to the SGD, the Mouse Genomic Informatics[34] (MGI) is a database resource that provides integrated genetic, genomic and biological data related to the Mus musculus (mouse). MGI includes a database (MGD) (Eppig, Bult et al. 2005) related to the genome information, such as characterization, nomenclature, mapping, gene homologies among mammals, sequence links, phenotypes, allelic variants and mutants, and strain data, as well as other databases that contains data of gene expression (GXD), tumor biology (MTB) and metabolism and cell level processes (MouseCyc). The fields "Name" and "Symbol" from this database have been used to construct the gene

[31] http://www.ncbi.nlm.nih.gov/taxonomy
[32] http://www.geneontology.org/
[33] http://www.yeastgenome.org/
[34] http://www.informatics.jax.org/

documents which are the basis of the disambiguation step (cf. 5.5) in the gene and protein normalization task.

A.7 FlyBase

Similar to SGD for the yeast, and MGI for the mouse, FlyBase[35] (Gelbart, Crosby et al. 1997) provides genetic and molecular information for the fly species, and specially for the Drosophila Melanogaster, the fruit fly, is one of the most studied eukaryotic organisms. The database includes information such as genes, alleles (and phenotypes), aberrations, transposons, pointers to sequence data, clones, stock lists, among others. The fields "Name", "Symbol", "Symbol Synonym" and "Name Synonym" from this database have been used to construct the gene documents which are the basis of the disambiguation step (cf. 5.5) in the gene and protein normalization task.

[35] http://flybase.org/

APPENDIX B: CORPORA

We have used many available corpora for the training and evaluation of the methodologies proposed in this thesis. In this section we give a brief description for each of them.

B.1 BioCreative II Gene Mention

The most widely used for gene/protein recognition dataset is the one made available in the BioCreative II Gene Mention Task[36] (Smith, Tanabe et al. 2008). It consists on 15,000 and 5,000 sentences for the training and testing datasets, respectively. The 15,000 training sentences were the same that were available in the BioCreative Task 1A (Yeh, Morgan et al. 2005) but a different dataset of 5,000 sentences were used for the blind test evaluation. The origins of the BioCreative Task 1A dataset was the GENETAG corpus (Tanabe, Xie et al. 2005) which was later combined with the MedTag (Smith, Tanabe et al. 2005) corpus, which also contains the ABGene corpus, to compose the GENETAG-5 corpus. A Perl script was available during the BioCreative II Gene Mention Task for the evaluation of the testing corpus or for a subset of the training corpus, in case of using a cross-validation approach. This was the corpus which has been used in the training and evaluation of the methodology we propose in section 4.6 for the extraction of genes and protein.

For the construction of this corpus, the sentences were selected at random from Medline, and half of them are likely to contain genes and proteins based on similarity to sentences with known gene names. The sentences were manually annotated by scientists with backgrounds in biochemistry, molecular biology and genetics manually. Genes and protein were annotated in such a way as to allow more than one alternative form, according to the boundaries of the mentions. An example of a sentence and annotations belonging to the corpus is shown below:

> P00089778A0000 The concentration of alpha 2-macroglobulin, alpha 1-antitrypsin, plasminogen, C3-complement, fibrinogen degradation products (FDP) and fibrinolytic activity, were studied in the aqueous humour and serum from nine patients with Fuchs' endothelial dystrophy, 17 patients with uncomplicated senile cataract and in the secondary aqueous from six cataract patients.

The genes which appear in the above text:

> P00089778A0000|18 37|alpha 2-macroglobulin
> P00089778A0000|39 56|alpha 1-antitrypsin
> P00089778A0000|58 68|plasminogen
> P00089778A0000|70 82|C3-complement
> P00089778A0000|84 112|fibrinogen degradation products
> P00089778A0000|114 116|FDP

[36] http://biocreative.sourceforge.net/biocreative_2_gm.html

And the alternative spanned text for the genes which appears in the above text:

P00089778A0000|70 71|C3
P00089778A0000|84 117|fibrinogen degradation products (FDP)
P00089778A0000|84 93|fibrinogen
P00100540T0000|52 66|cyclo-oxygenase

B.2 BioCreative Task 1B

There are few corpora available for the training and evaluation of gene and protein normalization. One of the most widely used corpus is the BioCreative Task 1B (Hirschman, Colosimo et al. 2005) provided in the first BioCreative challenge for the Saccharomyces cerevisiae (yeast), Mus musculus (mouse) and Drosophila melanogaster (fruit fly). The number of abstracts available for each of the datasets (training, development and test) and for each of the organisms is presented in Table B.1. The training and development sets correspond to the corpora used to train and test the participating systems, respectively, during development phase. An additional blind test with the test dataset was used for official evaluation. These are the corpora that have been used in the evaluation of the methodology we propose in Chapter 5 for the normalization of genes and proteins.

As an example of the data provided by this corpus, we present the abstract below identified as "yeast_00007_training" in the yeast training and its respective set of annotations:

> In the yeast S. cerevisiae, ARS (autonomously replicating sequence) elements located in the intergenic spacers of the rRNA gene locus are infrequently activated as origins of replication. We analyzed the rARS activation with a combination of neutral/neutral (N/N) two-dimensional (2D) gel electrophoresis and either the intercalating drug psoralen, which in vivo specifically marks the transcribing gene copies, or the selective accessibility of restriction sites in transcriptionally active genes. We found that initiation of replication starts at those rARSs placed immediately downstream of transcribing rRNA genes. This correlation between transcription and replication is consistent with the presence of nucleosome-free enhancers at each transcriptionally active gene copy and suggests that the transcription factor Abf1p is involved in replication initiation at the ARS in the rDNA gene locus.

And the identifiers of the gene and proteins:

yeast_00007_training	S0000277	Y	yeast_00007_training	S0004022	Y
yeast_00007_training	S0000897	Y	yeast_00007_training	S0004494	Y
yeast_00007_training	S0002411	Y	yeast_00007_training	S0004712	N
yeast_00007_training	S0002483	Y	yeast_00007_training	S0004837	Y
yeast_00007_training	S0002635	Y	yeast_00007_training	S0004897	N
yeast_00007_training	S0002777	Y	yeast_00007_training	S0005194	Y
yeast_00007_training	S0003131	Y	yeast_00007_training	S0005559	Y
yeast_00007_training	S0003490	Y	yeast_00007_training	S0005943	Y
yeast_00007_training	S0003628	Y			

The second column of the above example stands for the gene or protein identification in the SGD database (cf. A.5) and the third column indicates whether the mention is found in the abstract of the document or only in its full text. The mentions are not provided, but only its identifier according to the SGD, MGI (cf. A.6) and FlyBase (cf. A.7) databases for the yeast, mouse and fly organisms, respectively.

Corpora / Organism	**Yeast**	**Mouse**	**Fly**
Training	5,000	5,000	5,000
Development	110	250	108
Test	250	250	250

Table B.1: Details of the BioCreative Task 1B corpus.
The number of documents for the gene/protein normalization task for training, development and test corpora provided by BioCreative 1 task 1B (yeast, mouse and fly) are shown.

Besides the corpus, for each organism, a list of gene and protein synonyms has been provided which could be used in the construction of the normalization systems as a dictionary of synonyms or enriched with extra manually or automatically generated synonyms. This list contains the following number of synonyms for each of the organism: 14,995 for yeast, 130,208 for mouse and 116,744 for fly. We have used this list as basis for our gene/protein normalization methods (cf. 5.1).

B.3 BioCreative 2 Gene Normalization Task

The other widely used corpus for the gene/protein normalization task is the one for the Homo sapiens (human) which was provided during the BioCreative 2 Gene Normalization task (Morgan, Lu et al. 2008). The corpus is composed of 281 and 262 documents (title and abstracts) for the training and testing datasets, respectively. The gene/protein annotations are provided with their respective mention and are identified according to the Entrez Gene database (cf. A.3).

As example of this corpus, the document 8890164 and the annotated Entrez Gene entities are shown below. More than one alternative mention may be found for the entity. Although the mentions are provided, their exact localization in the text is not given.

> Association of inhibitory tyrosine protein kinase p50csk with protein tyrosine phosphatase PEP in T cells and other hemopoietic cells.
>
> p50csk is a tyrosine protein kinase (TPK) that represses the activity of Src family TPKs. We previously showed that Csk is a potent negative regulator of antigen receptor signaling in T lymphocytes and that its Src homology (SH) 3 and SH2 domains are required to inhibit these signals. To test the idea that the Csk SH3 and SH2 domains mediate interactions with other cellular proteins, we attempted to identify Csk-associated polypeptides using the yeast two-hybrid system. The results of our experiments demonstrated that Csk physically associates with PEP, a protein tyrosine phosphatase (PTP) expressed in hemopoietic cells. Further analyses revealed that this interaction was mediated by the Csk SH3 domain and by a proline-rich region (PPPLPERTP) in the non-catalytic C-terminal portion of PEP. The association between Csk and PEP was documented in transiently transfected Cos-1 cells and in a variety of cells of hemopoietic lineages, including T cells. Additional analyses demonstrated that the association between Csk and PEP is highly specific. Together, these data indicated that PEP may be an effector and/or a regulator of p50csk in T cells and other hemopoietic cells. Moreover, they allowed the identification of PEP as the first known ligand for the Csk SH3 domain.

And the annotations:

```
8890164  1445   inhibitory tyrosine protein kinase p50csk        p50csk  Csk
8890164  26191  PEP      protein tyrosine phosphatase PEP
```

Similar to the BioCreative Task 1B, the data provided by the challenge included a list of the synonyms for the human gene, which contained a total of 203,077 synonyms. This list has also been used as basis for our methods (cf. 5.1).

B.4 GENIA event

The GENIA corpus (Kim, Ohta et al. 2008) is also a very popular corpus in the biomedical text mining community. The GENIA event corpus consists of 1,000 PubMed abstracts which contain more than 9,000 sentences in which more than 36,000 events have been annotated. The annotated events allow a wide range of participants which are annotated according to predefines roles, such as theme, cause, site, location, etc. A part of the publicly available portion of this corpus has been used as training and development test datasets for the BioNLP Shared Task on Event Extraction challenge[37] (Kim, Ohta et al. 2009) while a held-out portion has been used for the blind test evaluation.

The corpus allows the identification of nine types of biological events: localization, binding, gene expression, transcription, protein catabolism, phosphorylation, regulation, positive regulation and negative regulation. It contains 800, 150 and 260 documents (title and abstract text only) on the biomedical domain that has been made available for the training, development test and blind test datasets, respectively. For all documents, the text comes previously annotated with the genes and proteins that might take part in the referred events.

[37] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/

Arguments	gene expression, transcription, protein catabolism	phosphorylation	localization	binding	regulation, postive regulation, negative regulation
Theme	✓	✓	✓	✓	✓*
Theme2			×		
Theme3			×		
Site		×	×		×
Site2			×		
Site3			×		
Cause					×*
CSite					×
AtLoc			×		
ToLoc			×		

Table B.2: Summary of the arguments for each of the biological events.
Mandatory arguments are marked with a check (✓) and the optional ones with a cross (×). For the regulatory events, the theme and cause arguments may be represented by a protein or by another event, marked with an asterix (*).

The main component of an event is the trigger token (or tokens) that indicate the change in the state of the bio-molecule, usually a verb. In addition, it may be composed of one or more arguments, depending on its type. The theme, usually a gene or a protein, is the only argument that is common to all types of events and it represents the main entity that takes part in the event. A summary of the arguments that each type of event may have is presented in Table B.2.

Figure B.1 shows examples for some types of events. Sentence 1 shows an example of a simple event of the gene expression type that has the token “expression” as its event trigger and the token “RANTES” as its only argument, the theme. Similar to the gene expression, transcription and protein catabolism events are also only associated to one theme argument. An example of a more complex event is also present in sentence 1, a positive regulation event that has the token “activates” as event trigger, the gene expression event as theme and a protein as cause. Regulatory events (regulation, positive regulation and negative regulation) are much more complex due to the fact that they may have an extra argument, a cause. Also, both the theme and cause arguments may be mapped to one of the previously annotated proteins or to any other event that comes before or after the referred regulatory event in the text.

Still in Figure B.1, sentence 2 shows another example of a negative regulation event as well as a phosphorylation event, which is associated to the protein “STAT1” and to the site “tyrosine”. The phosphorylation and localization events optionally may have a site or location associated to them and the latter does not come already annotated in the text. In contrast to the given proteins, they must be previously extracted from the text. An interesting point in this example is the token “expression” that it is not associated to a gene expression event and to the protein “interferon-gamma” protein, as it is not

annotated in the training document. An automatic solution to the event extraction problem should be able to deal with this type of situation.

1) *"RFLAT-1: a new zinc finger transcription factor that **activates** RANTES gene **expression** in T lymphocytes."* (document 10023774)

E1	Positive_regulation: **activates** Theme:E2 Cause: **RFLAT-1**
E2	Gene_expression: **expression** Theme: **RANTES**

2) *"Interleukin-10 inhibits expression of both interferon alpha- and interferon gamma- induced genes by **suppressing tyrosine phosphorylation** of STAT1."* (document 10029571)

E1	Negative_regulation: **supressing** Theme:E2 Cause: **Interleukin-10**
E2	Phosphorylation: **phosphorylation** Theme: **STAT1**. Site: **tyrosine**

3) *"When we analyzed the nature of STAT proteins capable of **binding** to IL-2Ralpha, pim-1, and IRF-1 **GAS elements** after cytokine stimulation, we observed IFN-alpha-**induced binding** of STAT1, STAT3, and STAT4, but not STAT5 to all of these elements."* (document 10068671)

E16	Binding: **binding** Theme: **pim-1** Site: **GAS elements**
M4	**Speculation** E16
E17	Binding: **binding** Theme: **IL-2Ralpha** Site: **GAS elements**
M5	**Speculation** E17
...	
E19	Positive_regulation: **induced** Theme:E33
E20	Positive_regulation: **induced** Theme:E35
...	
E31	Binding: **binding** Theme: **STAT4** Theme2: **IRF-1** Site2: **GAS elements**
E32	Binding: **binding** Theme: **STAT3** Theme2: **IL-2Ralpha** Site2: **GAS elements**
E33	Binding: **binding** Theme: **STAT3**Theme2: **IRF-1** Site2: **GAS elements**
E34	Binding: **binding** Theme: **STAT4** Theme2: **pim-1** Site2: **GAS elements**
E35	Binding: **binding** Theme: **STAT1** Theme2: **IL-2Ralpha** Site2: **GAS elements**
...	

Figure B.1: Examples of the GENIA event corpus.
In the examples above, the trigger events and other entities, such as sites, are shown in bold and the proteins are underlined.

Finally, sentence 3 shows some examples of the binding event, which may be considered the most complex of the events as it may be associated to two or more themes (proteins only) and up to one site for each theme. Once again, the sites should be previously extracted from the text. In this sentence, 17 events have been annotated using only 3 event triggers (two binding and one positive regulation), 6 proteins (shown underlined) and the site token "GAS elements". For some of these events, the referred site is associated to the main theme (cf. events E16, E17, E18, E37, E38, and E41) and in some other to the secondary theme (cf. other binding events). The use of a high level natural language processing is needed to correctly extract all the above events. Also, a third type of annotation is shown in this sentence, the modifiers of the event, in cases of speculation or negation, as it is the case of the annotations M4 and M5 for the events E16 and E17, respectively, due to the doubt in the expression "STAT proteins capable of binding to". The modifiers may be of speculation or negation only.

Three tasks have been proposed in the BioNLP'09 Event Extraction challenge[38] (cf. Figure B.2). Task 1 required the identification of the events (including their trigger tokens) and the association to the respective theme. Task 2 is an extension of Task 1 in which extra arguments are required, such as the cause for the regulatory events or the

[38] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/index.shtml#task

site or location argument from the binding, localization or phosphorylation events, that should be first extracted from the text, as they are not given. Task 3 is an extension of Task 2 and includes the annotation of the speculation and negation modifiers.

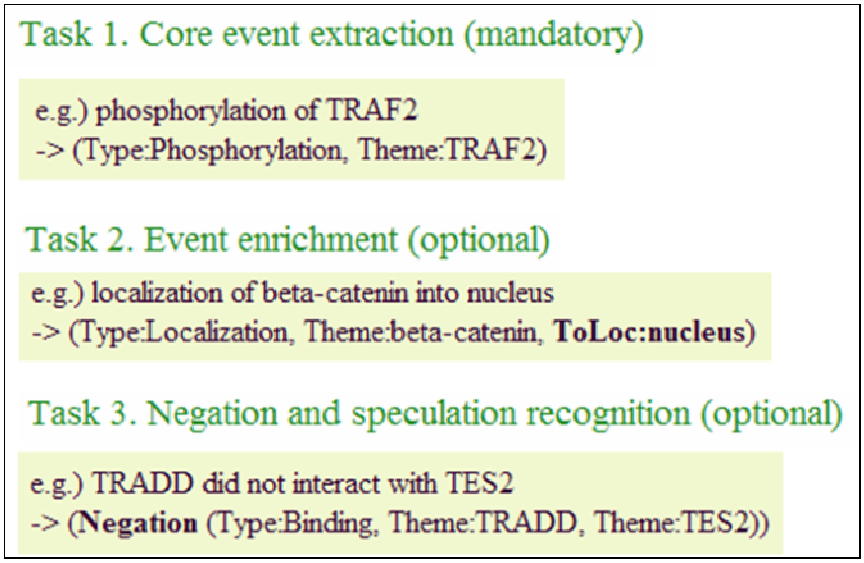


Figure B.2: Examples for each of the tasks.
Examples are shown for tasks 1, 2 and 3 from the BioNLP web site.

The evaluation for these tasks may be carried out for the development dataset using specific scripts[39] made available by the shared task organization and online[40] for the blind test dataset. More details on the corpus on the BioNLP'09 Event Extraction page.

B.5 BioText

The BioText[41] (Rosario and Hearst 2004) corpus is composed of 3655 sentences extracted from Medline which have been annotated with diseases and treatments entities. It is composed by the first 100 titles and the first 40 abstracts from the 59 files medline01n*.xml of 2001 version. It includes a total of almost 3,500 relationships between diseases and treatments. Each sentence has been annotated with the type of relationship between the disease and treatment, as for example, whether the treatment prevents or is effective (or not) to a certain disease, whether some side effects may be observed when using the specified treatment, or whether the association between both entities is vague. Also, some sentences are annotated with disease or treatment entities only. These sentences clearly do not show any relationship between those entities, as neither do the sentences that are not annotated with any type of entity at all. The types of relationships are described below:

[39] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/downloads.shtml
[40] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/eval-test.shtml
[41] http://biotext.berkeley.edu/data/dis_treat_data.html

- PREVENT: the treatment prevents the disease;
- SIDE_EFF: the treatment generates some side-effects;
- VAGUE: it is not clear if the treatment is effective for the disease;
- TREAT_ FOR_DIS: the treatment is effective for the disease;
- TREAT_NO_FOR_DIS: the treatment is not effective for the disease;
- NONE: no disease or treatment is present in the text.
- DISONLY: only the disease is present in the text, there is no relationship with any treatment;
- TREATONLY: only the treatment is present in the text, there is no relationship with any disease.

Relationships	Number of Sentences	Types of entities	Evaluation
NONE	1818	None	
TO_SEE	75	Diseases and Treatments	
DISONLY	629	Diseases	
TREATONLY	169	Treatments	
PREVENT	63	Diseases and Treatments	✓
SIDE_EFF	30	Diseases and Treatments	✓
VAGUE	37	Diseases and Treatments	✓
TREAT_FOR_DIS	830	Diseases and Treatments	✓
TREAT_NO_FOR_DIS	4	Diseases and Treatments	✓

Table B.3: Details of the BioText corpus.
The number of sentences and the types of entities that have been annotated for each type of relationship in the BioText corpus are shown. The relationships identified with a check mark were the ones that have been used in our evaluation.

Table B.3 shows the number of sentences annotated for each type of relationship. This corpus has been used in the evaluation of the relation extraction methodology proposed in 4.5. The table clearly shows that the corpus is very unbalanced and that the number of sentences annotated as "TREAT_NO_FOR_DIS" is extremely low.

Below we show some examples of the BioText corpus, for each of the categories under consideration in this thesis:

> CONCLUSIONS : In men and women 65 years of age or older who are living in the community , <TREAT_PREV> dietary supplementation with calcium and vitamin D </TREAT_PREV> moderately reduced <DIS_PREV> bone loss </DIS_PREV> measured in the femoral neck , spine , and total body over the three-year study period and reduced the incidence of <DIS_PREV> nonvertebral fractures </DIS_PREV> . || **PREVENT**

> Appetite suppressants-most commonly <TREAT_SIDE_EFF> fenfluramines </TREAT_SIDE_EFF> -increase the risk of developing <DIS_SIDE_EFF> PPH </DIS_SIDE_EFF> (odds ratio , 6.3) , particularly when used for more than 3 months (odds ratio , > 20) . || **SIDE_EFF**

Although long-term survival can be achieved by successful <TREAT_VAG> corrective surgery </TREAT_VAG> , the associated structural defects such as <DIS_VAG> large meningomyelocele </DIS_VAG> and severe <DIS_VAG> limb aplasia </DIS_VAG> or <DIS_VAG> hypoplasia </DIS_VAG> , as seen in our patient , can influence the patient 's quality of life . || **VAGUE**

CONCLUSION : <TREAT> Methylphenidate </TREAT> is effective in treating children with <DIS> epilepsy </DIS> and <DIS> ADHD </DIS> and safe in children who are seizure free . || **TREAT_FOR_DIS**

More of those initially prescribed <TREAT_NO> antibiotics </TREAT_NO> initially returned to the surgery with <DIS_NO> sore throat </DIS_NO> (38 % v 27 % , adjusted hazard ratio for return 1.39 % , 95 % confidence interval 1.03 to 1.89) . || **TREAT_NO_FOR_DIS**

APPENDIX C: AVAILABLE SOFTWARES

C.1 LingPipe

LingPipe is a tool that provides some computational linguistics services for the text processing. The functionalities go from simple tasks such as sentence detection and part-of-speech tagging to more complex ones such as named-entity recognition. LingPipe Java library[42] is freely available as well as the models which are necessary to some of the services. In this thesis, LingPipe has been widely used for sentence detection, tokenization and part-of-speech tagging, as for example, in the relationship extraction tasks (cf. 4.3).

C.2 Porter Stemmer

The Porter stemming algorithm (Porter 1980) is most used stemmer in the natural language field. It removes morphological and inflexional endings from words for the English language. For example, the words "connect", "connected", "connecting", "connection" and "connections" and all mapped to the stem "connect". The algorithm consists basically of a set of rules that map the words to their stem according to the endings. In this thesis, a freely available Java implementation[43] of this algorithm has been used. In this thesis, it has been used in the disambiguation step of the gene and protein normalization task (cf. 5.5).

C.3 Dragon Toolkit

The Dragon Toolkit[44] (Zhou, Zhang et al. 2007) is a freely available Java library which provides functionalities related to the information retrieval and text mining domains. These are some of the features included in it: sparse matrix representation, document representation, text clustering, text classification, text summarization, etc. In this thesis, only the lemmatizer (cf. 3.1.5) included in this library has been used in the relationship extraction tasks (cf. 4.3).

C.4 Stanford parser

The Standford parser[45] (Klein and Manning 2003) is a Java implementation of probabilistic natural language parsers (cf. 3.1.6), both highly optimized PCFG (Probabilistic Context Free Grammar) and lexicalized dependency parsers, and a lexicalized PCFG parser. The Stanford parser has been widely used in this thesis as some of the features that are used in the case-based reasoning approach for the

[42] http://alias-i.com/lingpipe/index.html
[43] http://tartarus.org/~martin/PorterStemmer/
[44] http://dragon.ischool.drexel.edu/
[45] http://nlp.stanford.edu/software/lex-parser.shtml

relationship extraction tasks (cf. 4.3.2), are related to the syntactic structure of the sentence.

C.5 ABNER

ABNER[46] (Settles 2005) is a gene and protein recognition tool which is available online through a web interface or as a Java library. It is one of the most used gene and protein tagger in the biomedical community. The core of ABNER is a statistical machine learning system using linear-chain conditional random fields with a variety of orthographic and contextual features. It includes two models trained on the NLPBA and BioCreative corpora with an f-measure performance of 70.5 and 69.9, respectively. ABNER has been used in this thesis for comparison to proposed gene and protein recognition task (cf. 4.6) and as part of the mix of tagger used in the gene and protein normalization tests (cf. Chapter 5).

C.6 BANNER

BANNER[47] (Leaman and Gonzalez 2008) is a biomedical named entity recognition system. It consists of a machine-learning system based on conditional random fields and uses the best features in recent literature on biomedical named entity recognition. BANNER has been trained and evaluated on the BioCreative II Gene Mention corpus (cf. B.1) and it has obtained an f-measure of 84.92. Similar to ABNER, BANNER has been used in this thesis for comparison to proposed gene and protein recognition task (cf. 4.6)and as part of the mix of tagger used in the gene and protein normalization tests (cf. Chapter 5).

[46] http://pages.cs.wisc.edu/~bsettles/abner/
[47] http://cbioc.eas.asu.edu/banner/

APPENDIX D: ADDITIONAL TABLES

D.1 Global Alignment costs for the comparison of part-of-speech tags

Chunk	np	term	link	be	aux	adj	adv	cnj	det	prep	pron	pun	verb	exc
np	0.20	0.30	0.90	0.90	0.90	0.75	0.90	1.00	0.90	1.00	0.15	0.90	0.70	0.90
term	0.30	0.15	1.00	0.90	0.90	0.80	0.95	1.00	1.00	1.00	0.15	1.00	0.70	1.00
link	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
be	0.90	0.90	0.40	0.00	0.10	0.90	0.75	0.55	0.40	0.70	0.90	0.40	0.55	0.40
aux	0.90	0.90	0.40	0.10	0.00	0.90	0.75	0.55	0.40	0.70	0.90	0.40	0.55	0.40
adj	0.75	0.80	0.50	0.90	0.90	0.15	0.25	0.65	0.50	0.85	0.75	0.50	0.90	0.50
adv	0.90	0.95	0.50	0.75	0.75	0.25	0.15	0.65	0.50	0.85	0.90	0.50	0.80	0.50
cnj	1.00	1.00	0.20	0.55	0.55	0.65	0.65	0.00	0.20	0.55	1.00	0.20	0.95	0.20
det	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
prep	1.00	1.00	0.50	0.70	0.70	0.85	0.85	0.55	0.50	0.05	1.00	0.50	1.00	0.50
pron	0.15	0.15	0.85	0.90	0.90	0.75	0.90	1.00	0.85	1.00	0.05	0.85	0.70	0.90
pun	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.85	0.00	0.90	0.00
v	0.70	0.70	0.90	0.55	0.55	0.90	0.80	0.95	0.90	1.00	0.70	0.90	0.20	0.90
exc	0.90	1.00	0.00	0.40	0.40	0.50	0.50	0.20	0.00	0.50	0.90	0.00	0.90	-

Table D.1: Costs for the global alignment of the part-of speech tag in case comparison.
Original costs proposed by (Spasic, Ananiadou et al. 2005). Abbreviation: np (noun phrase), be (to be verb), aux (auxiliary), adj (adjective), adv (adverb), cnj (conjunction), det (determiners), prep (preposition), pun (punctuation), exc (exclusion).

APPENDIX E: ADDITIONAL RESOURCES

E.1 List of stopwords

a	both	four
about	bottom	from
above	but	front
across	by	full
after	call	further
afterwards	can	get
again	cannot	give
against	cant	go
all	co	had
almost	computer	has
alone	con	hasnt
along	could	have
	couldnt	he
also	cry	hence
although	de	her
always	describe	here
am	detail	hereafter
among	do	hereby
amongst	done	herein
amoungst	down	hereupon
amount	due	hers
an	during	herself
and	each	him
another	eg	himself
any	eight	his
anyhow	either	how
anyone	eleven	however
anything	else	hundred
anyway	elsewhere	i
anywhere	empty	ie
are	enough	if
around	etc	in
as	even	inc
at	ever	indeed
back	every	interest
be	everyone	into
became	everything	is
because	everywhere	it
become	except	its
becomes	few	itself
becoming	fifteen	keep
been	fify	last
before	fill	latter
beforehand	find	latterly
behind	fire	least
being	first	less
below	five	ltd
beside	for	made
besides	former	many
between	formerly	may
beyond	forty	me
bill	found	meanwhile

might	serious	un
mill	several	under
mine	she	until
more	should	up
moreover	show	upon
most	side	us
mostly	since	very
move	sincere	via
much	six	was
must	sixty	we
my	so	well
myself	some	were
name	somehow	what
namely	someone	whatever
neither	something	when
never	sometime	whence
nevertheless	sometimes	whenever
next	somewhere	where
nine	still	whereafter
no	such	whereas
nobody	system	whereby
none	take	wherein
noone	ten	whereupon
nor	than	wherever
not	that	whether
nothing	the	which
now	their	while
nowhere	them	whither
of	themselves	who
off	then	whoever
often	thence	whole
on	there	whom
once	thereafter	whose
one	thereby	why
only	therefore	will
onto	therein	with
or	thereupon	within
other	these	without
others	they	would
otherwise	thick	yet
our	thin	you
ours	third	your
ourselves	this	yours
out	those	yourself
over	though	yourselves
own	three	
part	through	
per	throughout	
perhaps	thru	
please	thus	
put	to	
rather	together	
re	too	
same	top	
see	toward	
seem	towards	
seemed	twelve	
seeming	twenty	
seems	two	

APPENDIX F: SOFTWARE DEVELOPED

In this section we describe the software which has been developed during the thesis and which implements some of the methodologies proposed here.

F.1 Moara Project

(http://moara.dacya.ucm.es)

The Moara project is a Java library oriented to gene and protein recognition and normalization tasks, carried out by the systems CBR-Tagger and ML-Normalization, respectively. Moara makes use of some MySQL databases and three external libraries: Weka machine learning tool[48], SecondString library[49] for string distance metrics, and ABNER (cf. C.5) as an additional tagger for the extraction of gene and protein mentions. Moara is available through Sourceforge[50] as well as integrated into the U-Compare[51] framework (Kano, Baumgartner et al. 2009) as a UIMA component (Baumgartner, Lu et al. 2008). A comparison of Moara with other similar systems is shown in Table F.1 at the end of this section. The MySQL databases store data that have been learned by the system during the training phases as well as the external data that are necessary for some of the functionalities of the system. The four databases in Moara are listed below:

- moara: contains general and biological data that are of use for the functionalities in the project. This database holds the data related to stopwords[52], Biothesaurus (cf. A.2) biomedical terms and a list of all organisms present in Entrez Gene Taxonomy (cf. A.3). This database is essential for all functionalities of the Moara project.

- moara_mention: contains data (cases) which were learned during the training step of CBR-Tagger; it is used for extracting gene and protein mentions from texts.

- moara_gene: contains data related to the genome, and a dictionary of synonyms of the organisms under consideration. The current version supports yeast, mouse, fly and human. This data are used for both the matching procedure and the disambiguation strategy of the gene and protein normalization task.

- moara_normalization: contains data related to the transformations that have been applied to the gene/protein synonyms in order to compose the features that take part in the machine learning matching procedure of the normalization task.

[48] http://www.cs.waikato.ac.nz/ml/weka/
[49] http://secondstring.sourceforge.net/
[50] http://sourceforge.net/projects/moara/
[51] http://u-compare.org/components/components-semantic_tools.html
[52] http://moara.dacya.ucm.es/download.html

The functionalities available for the CBR-Tagger and the ML-Normalization systems are described in details in the following pages. Examples of code are provided for each of them.

F.1.1 CBR-Tagger

Gene and protein recognition is carried out by the CBR-Tagger (Neves, Carazo et al. 2010), a tagger based on Cased-based reasoning (CBR) foundations (cf. 4.1). Besides extracting mentions from a text, it is also possible to train the CBR-Tagger with different documents. In addition, a wrapper of the ABNER tagger was developed in order to use its mentions without the need to learn the ABNER library. The methodology behind the CBR-Tagger has been described in section 4.6 of this work. The functionalities available in CBR-Tagger are listed below.

Extraction of mentions with CBR-Tagger

There are five built-in models in the "moara_mention" database. CBR-Tagger has been trained with the training set of documents made available during the BioCreative 2 Gene Mention task (cf. B.1) and with additional corpora to improve the extraction of mentions from different organisms. These extra corpora belong to the gene normalization datasets for the BioCreative task 1B (cf. B.2) corresponding to yeast, mouse and fly gene/protein normalization. These training datasets are referred as CbrBC2, CbrBC2y, CbrBC2m, CbrBC2f and CbrBC2ymf, depending if they are composed by the BioCreative 2 Gene Mention task corpus alone or combined with the BioCreative task 1B corpus for the yeast, mouse, fly or all three, respectively. Five constants are available to refer to each of these models.

There is no requirement to retrain the system; all these models are included by default in the specified database. The extraction method receives two string arguments: the predefined or user-specific model used to train the tagger and the text from which the mention are to be recognized. Two version of the CBR-Tagger were integrated into U-Compare framework[53] using the models CbrBC2 and CbrBC2ymf.

We show an example below in which genes and proteins are extracted from a short text. The CbrBC2 model is used and the method "extract" of the "GeneRecognition" class is used. The "GeneMention" object encapsulates a gene or protein and provides means to access its attributes, such as the text of the mention and the start and end character of it in the text.

[53] http://u-compare.org/components/components-semantic_tools.html

```
import moara.mention.MentionConstant;
import moara.mention.functions.GeneRecognition;
import moara.mention.entities.GeneMention;
import java.util.ArrayList;

public class TestExtraction {

   public static void main(String[] args) {

      // Abstract Pubmed Id 8076837
      String text = "A gene (pkt1) was isolated from the " +
        "filamentous fungus Trichoderma reesei, which " +
        "exhibits high homology with the yeast YPK1 and " +
        "YKR2 (YPK2) genes. It contains a 2123-bp ORF " +
        "that is interrupted by two introns, and it " +
        "encodes a 662-amino-acid protein with a " +
        "calculated M(r) of 72,820. During active growth, " +
        "pkt1 is expressed as two mRNAs of 3.1 and 2.8 kb " +
        "which differ in the 3' untranslated region due to " +
        "the use of two different polyadenylation sites. ";
      // Extracting...
      GeneRecognition gr = new GeneRecognition();
      ArrayList<GeneMention> gms =
        gr.extract(MentionConstant.MODEL_BC2,text);
      // Listing mentions...
      System.out.println("Start\tEnd\tMention");
      for (int i=0; i<gms.size(); i++) {
         GeneMention gm = gms.get(i);
         System.out.println(gm.Start() + "\t" + gm.End() +
           "\t" + gm.Text());
      }

   }

}
```

And below is the output for the code above, showing the offsets of the genes and protein found in the text provided.

```
Start End   Mention
96    104   yeast YPK1
108   122   YKR2(YPK2) genes
```

Extraction of mentions with ABNER

We have developed a wrapper for the ABNER tagger in order to allow a mix of taggers to be used when extracting mentions, with no need to learn the details of an extra library. ABNER comes with two models based on the corpora of the NLPBA[54] and BioCreative task 1A challenges. We have constructed five more models for ABNER, namely AbnerBC2, AbnerBC2y, AbnerBC2m, AbnerBC2f and AbnerBC2ymf, by training it with the same datasets that were used for CBR-Tagger.

```
import java.util.ArrayList;

import moara.mention.entities.GeneMention;
import moara.wrapper.WrapperConstant;
import moara.wrapper.abner.AbnerTagger;

public class TestWrapper {

   public static void main(String[] args) {

      // Abstract Pubmed Id 8076837
      String text = "A gene (pkt1) was isolated from the " +
        "filamentous fungus Trichoderma reesei, which " +
        "exhibits high homology with the yeast YPK1 and " +
        "YKR2 (YPK2) genes. It contains a 2123-bp ORF " +
        "that is interrupted by two introns, and it " +
        "encodes a 662-amino-acid protein with a " +
        "calculated M(r) of 72,820. During active growth, " +
        "pkt1 is expressed as two mRNAs of 3.1 and 2.8 kb " +
        "which differ in the 3' untranslated region due to " +
        "the use of two different polyadenylation sites. ";
      // Extracting...
      AbnerTagger abner =
        new AbnerTagger(WrapperConstant.ABNER_BC2);
      ArrayList<GeneMention> gms = abner.extract(text);
      // Listing mentions...
      System.out.println("Start\tEnd\tMention");
      for (int i=0; i<gms.size(); i++) {
         GeneMention gm = gms.get(i);
         System.out.println(gm.Start() + "\t" + gm.End() +
           "\t" + gm.Text() + "\t");
      }

   }

}
```

The code above extracts genes for the same text of the example with the CBR-Tagger. Now the extraction is perfoemd with ABNER trained on the model AbnerBC2.

[54] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html

And below we present the output for the code above, showing the output of the ABNER tagger and the genes and proteins which have been extracted. They are not exactly the same as the ones extracted using CBR-Tagger.

```
Loading external tagging module from
'F:\Mariana\workspace\Moara\wrappers\abner\abner_bc2.model'...
Logging configuration class
"edu.umass.cs.mallet.base.util.Logger.DefaultConfigurator" failed
java.lang.ClassNotFoundException:
edu.umass.cs.mallet.base.util.Logger.DefaultConfigurator
Start End   Mention
6     9     pkt1
101   104   YPK1
108   111   YKR2
113   116   YPK2
256   259   pkt1
```

Training the CBR-Tagger

CBR-Tagger can be trained with extra corpora if the documents are provided in the format used in the BioCreative 2 Gene Mention task (cf. B.1), in which the text of the documents and the annotated gene and protein mentions are provided in two distinct files. Additionally, cases that have been learned for CBR-Tagger beforehand, from the aforementioned five training datasets (CbrBC2, CbrBC2y, CbrBC2m, CbrBC2f and CbrBC2ymf), can also be considered. CBR-Tagger provides a method for using these cases automatically, without the need to train the tagger again for that corpus.

The code below shows an example of training CBR-Tagger with different documents. The function "useDataModel" allows using the cases learned for the CbrBC2f model. Additionally, the system is trained using the text and the annotations provided in the files "train.txt" and "annotations.txt".

```
import moara.mention.MentionConstant;
import moara.mention.functions.TrainTagger;

public class TestTrainTagger {

   public static void main(String[] args) {

      TrainTagger tt = new TrainTagger();
      tt.useDataModel(MentionConstant.MODEL_BC2F);
      tt.readDocuments("train.txt");
      tt.readAnnotations("annotations.txt");
      tt.train();

   }

}
```

F.1.2 ML-Normalization

The normalization task is carried out by ML-Normalization, which include a exact (cf. 5.2) and a machine learning matching (cf. 5.4) approaches as well as a disambiguation strategy based on the text under consideration (cf. 5.5). The system uses freely available minimum organism-specific data. This is especially useful if no specifically tailored dictionary is available. The normalization step was trained for the four organisms: yeast, mouse, fly and human. The functionalities available in ML-Normalization are described below.

Normalizing mentions by exact matching

This methodology is carried out by performing an exact matching between the mention extracted from the text and the synonyms in the dictionaries (cf. 5.4). Both the mention and the synonyms are previously edited by dividing and filtering the tokens according to punctuations, numbers, Greek letters, and BioThesaurus terms (cf. A.2), and finally ordering the parts of the token alphabetically.

The example below shows how to normalize gene and protein mentions for the yeast organism using ML-Normalization. The first part of the code is omitted due to space reason. Here the mene and protein mentions are extracted by any available system, for instance CBR-Tagger or ABNER. Then the extracted menton are presented to the ML-Normalization as an array of “GeneMention” objects. The normalization is perfomed by the “normalize” of the “ExactMatchingNormalization” class. The output of the ML-Normalization is an array of “GenePrediction” objects, which encapsulate its attributes, such as the identifier to which the mention has been normalized, the score of the matching, the synonyms to which the mention has been matched, etc. More than one synonym may have been matched, and the best solution can be returned by the “GeneId()” method of the “GeneMention” object.

```
import moara.mention.MentionConstant;
import moara.mention.functions.GeneRecognition;
import moara.mention.entities.GeneMention;
import moara.normalization.functions.ExactMatchingNormalization;
import moara.normalization.entities.GenePrediction;
import moara.util.Constant;
import moara.bio.entities.Organism;
import java.util.ArrayList;

public class TestMoara {

   public static void main(String[] args) {

      ....

      // Normalizing mentions...
      Organism yeast = new Organism(Constant.ORGANISM_YEAST);
      ExactMatchingNormalization gn =
        new ExactMatchingNormalization(yeast);
      gms = gn.normalize(text,gms);

      // Listing normalized identifiers...
      System.out.println("\nStart\tEnd\t#Pred\tMention");
      for (int i=0; i<gms.size(); i++) {
         GeneMention gm = gms.get(i);
         if (gm.GeneIds().size()>0) {
            System.out.println(gm.Start() + "\t" + gm.End() +
              "\t" + gm.GeneIds().size() + "\t" +
              text.substring(gm.Start(),gm.End);
            for (int j=0; j<gm.GeneIds().size(); j++) {
               GenePrediction gp = gm.GeneIds().get(j);
               System.out.print("\t" + gp.GeneId() + " " +
                 gp.OriginalSynonym() + " " + gp.ScoreDisambig());
               System.out.println(
                (gm.GeneId().GeneId().equals(gp.GeneId())?
                " (*)":""));
            }
        }
      }

   }

}
```

Below we show the output of the normalization of the genes showed in the code above. First the genes extracted by the tagger are listed, and then, for each of them, the candidates of the normalization are shown. For the first gene (yeast YPK1), 3 candidates have been found and the best one if the second one (marked with and asterisk). For the second gene, just one candidate for normalization was found.

```
Start End   Mention
96    104   yeast YPK1
108   122   YKR2(YPK2) genes

Start End   #Pred Mention
96    104   3     yeast YPK1
      S000005251 YPK1 0.08569736765753885
      S000001609 YPK1 0.8058689254584626 (*)
      S000003629 YPK1 0.05729855793043799
108   122   1     YKR2 (YPK2) genes
      S000004710 YKR2 0.0 (*)
```

Using the dictionary of synonyms

We have made available a specific function for editing a given mention using the same methods we propose for the gene/protein normalization (cf. 5.2). In this way, it is possible to use our edited dictionary of synonyms with other matching procedures. The function we made available receives the text of the mention as input and returns a list of its variations.

The code below generate variations for the mention "YPK1 and YKR2(YPK2) genes" for the yeast organism.

```
import java.util.ArrayList;

import moara.util.Constant;
import moara.normalization.functions.ExactMatchingNormalization;
import moara.bio.entities.Organism;

public class TestDictionarySynonyms {

   public static void main(String[] args) {

      Organism yeast = new Organism(Constant.ORGANISM_YEAST);
      ExactMatchingNormalization app = new
        ExactMatchingNormalization(yeast);
      ArrayList<String> variations =
        app.getFlexibleMentions("YPK1 and YKR2(YPK2) genes");
      for (int i=0; i<variations.size(); i++)
         System.out.println(variations.get(i));

   }

}
```

The variations returned by the system are printed below:

```
1 2 2 and genes ykr ypk ypk
1 2 2 genes ykr ypk ypk
1 2 2 ykr ypk ypk
2 ypk
1 2 and genes ykr ypk
1 2 genes ykr ypk
1 2 ykr ypk
1 ypk
2 ykr
```

Training of the exact matching normalization

The exact matching for the ML-Normalization is trained for using the exact matching strategy for four organisms: yeast, mouse, fly and human. However, new organisms may be added to the system by providing general available information such as the code of the specified organism in NCBI Taxonomy (cf. A.3). Minimum organism-specific information must be provided, the "gene_info.gz" and "gene2go.gz" files from Entrez Gene FTP[55], but no gene normalization class needs to be created.

The code below shows a code for training the system for normalization of gene and protein mentions for the Bos Taurus (cattle) organism. It is only necessary to assign a name for the organism (cattle) and provide its identifier in NCBI Taxonomy (9913).

```
import moara.normalization.functions.TrainNormalization;
import moara.bio.entities.Organism;

public class TestNewOrganism {

   public static void main(String[] args) {

      Organism cattle = new Organism("9913");
      String name = "cattle";
      String directory = "normalization";

      TrainNormalization tn = new TrainNormalization(cattle);
      tn.train(name,directory);

   }

}
```

[55] ftp://ftp.ncbi.nih.gov/gene/DATA/

Normalizing mentions by machine learning matching

In addition to flexible matching, an approximated machine learning matching is provided for the normalization procedure. The strategy is based on the methodology described in section 5.4. Listed below are the parameters that can be chosen when using machine learning matching for the gene/normalization task:

- Percentage similarity: any value between 0 and 1 (0.9 by default);
- Selection of the pair of mention-synonyms: bigram or trigram similarity, or both (default option);
- Machine learning algorithm: Support Vector Machines (default option), Random Forests or Logistic Regression;
- Set of pair-features: all of them (indicative of equal prefixes, suffixes, numbers and Greek letters, bigram/trigram similarity, string similarity and shape similarity) or just the best of them (bigram/trigram similarity, number and string similarity) (default option).
- String similarity method: Levenstein, Jaro-Winkler, Smith-Waterman (default option), Monge-Elkan or Soft-TFIDF.

The default values shown in the list of parameters above represent the configuration of the system that works reasonably well for the four organisms we have considered (yeast, mouse, fly and human). Therefore, Moara comes with four previously learned models using the default values, one for each of the organisms under consideration.

The code below show an example of normalizing gene/protein mentions for the yeast using the machine learning matching. We have omitted the extraction of the genes and protein, which can be performed by any tagger, and printing the output of the system, similar to the one listed for the exact matching above.

Ml-Normalization provides methods for set each one of the parameters above. When using a value different of the default ones, the system should be first trained, as described below in this thesis.

```
import moara.mention.MentionConstant;
import moara.mention.functions.GeneRecognition;
import moara.mention.entities.GeneMention;
import moara.normalization.functions.MachineLearningNormalization;
import moara.normalization.entities.GenePrediction;
import moara.util.text.StringUtil;
import moara.util.Constant;
import moara.bio.entities.Organism;
import java.util.ArrayList;

public class TestMLNormalization {

   public static void main(String[] args) {

      …

      // Normalizing mentions...
      Organism yeast = new Organism(Constant.ORGANISM_YEAST);
      MachineLearningNormalization gn =
         new MachineLearningNormalization(yeast);
      gms = gn.normalize(text,gms);

      // Listing normalized identifiers...
      …

   }

}
```

And below we show the output provided by the system. Once again, three candidates wer found for the first mention and just one for the second mention. The candidates marked with an asterisk are the best identifier choosn by the disambiguation strategy.

```
Start End   Mention
96    104   yeast YPK1
108   122   YKR2(YPK2) genes
Reading model...models/model_yeast_svm_sw_f2_09_both.model

Start End   #Pred Mention
96    104   3     yeast YPK1
      S000005251 YPK1 0.08569736765753885
      S000001609 YPK1 0.8058689254584626 (*)
      S000003629 YPK1 0.05729855793043799
108   122   1     YKR2 (YPK2) genes
      S000004710 YKR2 0.0 (*)
```

Training of the machine learning matching

Training the machine learning matching is possible for values of parameters different of the built-in models, as well as for new organisms. In the latter case, the procedure to be used is the same as the one presented for exact matching, with the exception that we must ask the system to generate data for the machine learning matching as well.

In order to normalize the mentions using a model based on parameters others than the default ones, the system must first be trained to create the specified model. This procedure can be time-consuming depending on the number of synonyms for the organism under consideration as well as the parameters that have been chosen.

The "MachineLearningModel" class provides functions for setting any of the parameters discussed above. Afer the training, the system would be ready for normalizing the mentions using the previously trained model. In order that the system uses the model under consideration rather than the default one, the parameters for the "MachineLearningNormalization" class must be explicitly specified, just as carried out for the "MachineLearningModel" class when training the system. The example below illustrates how to train the system for some parameter different from the default ones.

```
import moara.normalization.NormalizationConstant;
import moara.normalization.functions.MachineLearningModel;
import moara.util.Constant;
import moara.bio.entities.Organism;

public class TestTrainingNormalization {

   public static void main(String[] args) {

      Organism yeast = new Organism(Constant.ORGANISM_YEAST);

      MachineLearningModel mlm = new MachineLearningModel(yeast);
      mlm.setPctSymilarity(0.6);
      mlm.setFeatures(NormalizationConstant.NAME_FEATURES_F1);
      mlm.setStringSimilarity(Constant.DISTANCE_SMITH_WATERMAN);
      mlm.setMachineLearningAlgorithm(Constant.ML_SVM);
      mlm.setGramSelection(NormalizationConstant.FEATURE_BIGRAM);

      mlm.train();

   }

}
```

Disambiguation of identifiers

When more than one identifier is obtained for a mention, a disambiguation procedure is used to decide which is more likely to be correct. The methodology behind this functionality is described in section 5.5 of this work. The user may choose not to use any disambiguation functionality or three types of measures, based on cosine similarity, number of common words (default option) or a mix of both. Also, choosing between single (default option) and multiple disambiguation selection is possible at this step. The single option selects only the best candidate; the multiple selection returns the top scored ones according to a given threshold.

Both the exact matching (ExactMatchingNormalization class) and the machine learning matching (MachineLearningNormalization class) provide means to choose among the many disambiguation options.

F.2 Moara BioEvent Extractor in U-Compare

(http://u-compare.org/)

The methodology described in section 4.4.2 has been implemented using Java language and MySQL technology. It has been added as a web service into the BioNLP server which integrates through the U-Compare[56] framework (Kano, Baumgartner et al. 2009) some of the solutions that have been developed for the BioNLP Event Extraction Shared Task challenge (Kim, Ohta et al. 2009), the U-Compare Bio-Event Meta-Service (Kano, Bjorne et al. 2011). A screenshot of the U-Compare is showed in Figure F.1.

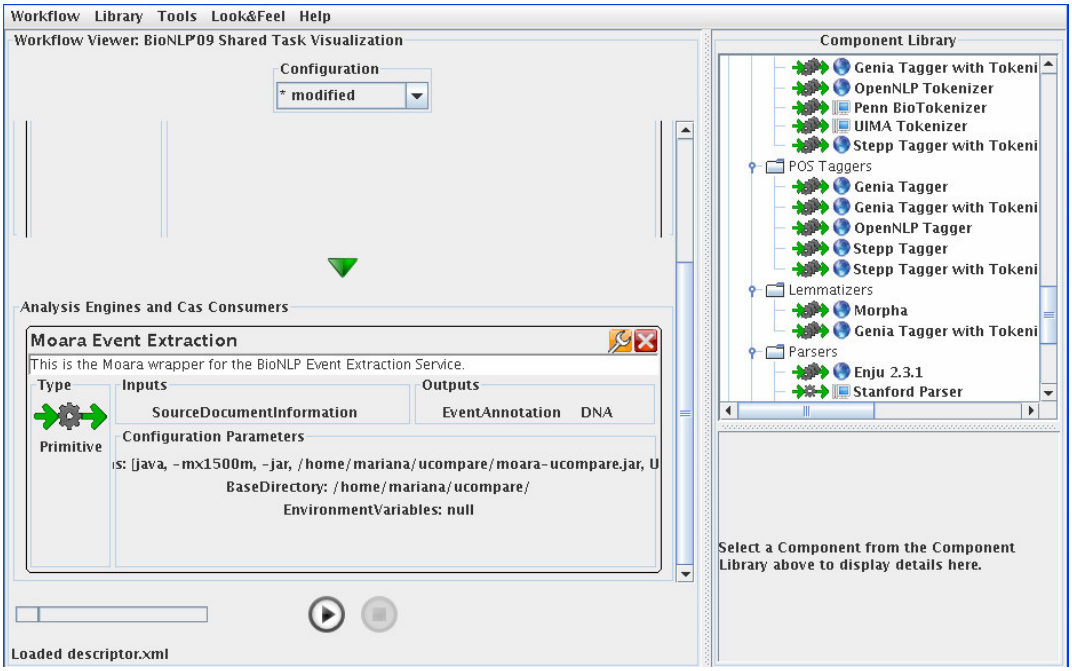
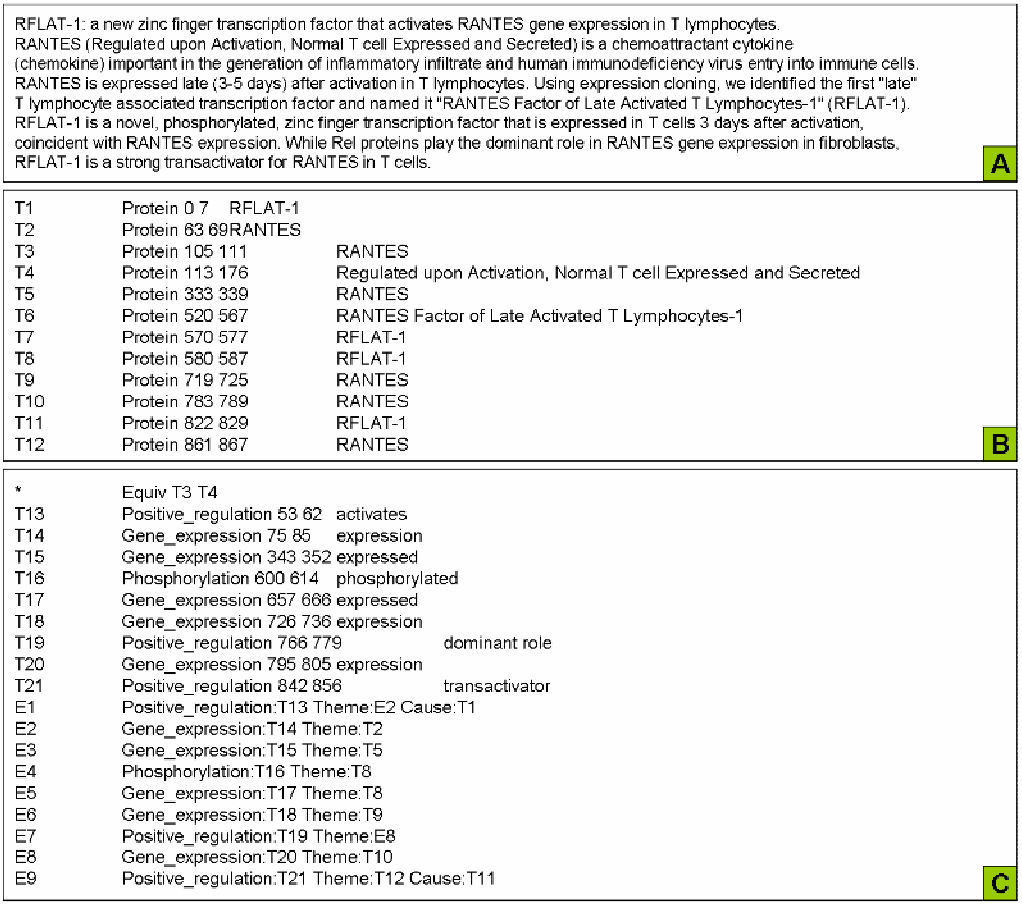


Figure F.1: Screenshot of the U-Compare system.
The main page of the U-Comapre system is shown. The components listed on the right may be used on the left for creating a pipeline for the extraction of events.

Moara BioEvent Extractor receives as input two documents, one composed of the free text to be processed by the system, usually a title and an abstract of a PubMed document (cf. A.1), and a second one which contains a list the proteins that have been found in the text and which might be involved in a biological event. The events that may be extracted from the text are: localization, binding, gene expression, transcription, protein catabolism, phosphorylation, regulation, positive regulation and negative regulation. Also, a modifier can be also associated to of the event, whether is a speculation or a

[56] http://u-compare.org/

negation, and may also be extracted from the text. More details on the GENIA event corpus can be found in section B.4. In the output file, the events and their respective arguments are listed according to the order of appearance in the text. Figure F.2 shows an example of the input and output files.



A

```
RFLAT-1: a new zinc finger transcription factor that activates RANTES gene expression in T lymphocytes.
RANTES (Regulated upon Activation, Normal T cell Expressed and Secreted) is a chemoattractant cytokine
(chemokine) important in the generation of inflammatory infiltrate and human immunodeficiency virus entry into immune cells.
RANTES is expressed late (3-5 days) after activation in T lymphocytes. Using expression cloning, we identified the first "late"
T lymphocyte associated transcription factor and named it "RANTES Factor of Late Activated T Lymphocytes-1" (RFLAT-1).
RFLAT-1 is a novel, phosphorylated, zinc finger transcription factor that is expressed in T cells 3 days after activation,
coincident with RANTES expression. While Rel proteins play the dominant role in RANTES gene expression in fibroblasts,
RFLAT-1 is a strong transactivator for RANTES in T cells.
```

B

```
T1      Protein 0 7    RFLAT-1
T2      Protein 63 69RANTES
T3      Protein 105 111        RANTES
T4      Protein 113 176        Regulated upon Activation, Normal T cell Expressed and Secreted
T5      Protein 333 339        RANTES
T6      Protein 520 567        RANTES Factor of Late Activated T Lymphocytes-1
T7      Protein 570 577        RFLAT-1
T8      Protein 580 587        RFLAT-1
T9      Protein 719 725        RANTES
T10     Protein 783 789        RANTES
T11     Protein 822 829        RFLAT-1
T12     Protein 861 867        RANTES
```

C

```
*       Equiv T3 T4
T13     Positive_regulation 53 62  activates
T14     Gene_expression 75 85      expression
T15     Gene_expression 343 352 expressed
T16     Phosphorylation 600 614    phosphorylated
T17     Gene_expression 657 666 expressed
T18     Gene_expression 726 736 expression
T19     Positive_regulation 766 779            dominant role
T20     Gene_expression 795 805 expression
T21     Positive_regulation 842 856            transactivator
E1      Positive_regulation:T13 Theme:E2 Cause:T1
E2      Gene_expression:T14 Theme:T2
E3      Gene_expression:T15 Theme:T5
E4      Phosphorylation:T16 Theme:T8
E5      Gene_expression:T17 Theme:T8
E6      Gene_expression:T18 Theme:T9
E7      Positive_regulation:T19 Theme:E8
E8      Gene_expression:T20 Theme:T10
E9      Positive_regulation:T21 Theme:T12 Cause:T11
```

Figure F.2: Example of the input and output files.
Example of the training document 10023774. A) .txt file: Title (first line) and abstract of the document. B) .a1 file: List of given proteins which have been identified in the text. C) .a2 file (output): List of the events and entities (sites, locations) which should be extracted from the text.

Moara BioEvent Extractor first separates the sentences and tokenizes them using functionalities included in the LingPipe library (cf. C.1). Then, it uses the Stanford parser (cf. C.4) for the generation of the syntactic structure and the dependency parser for each of the sentences. The system then proceeds to the processing of the document, i.e., the extraction of the named entities (event triggers, sites and locations) and the biological event, both based on case-based reasoning (CBR) classifiers, as described in section 4.4 of this thesis. One additional library was integrated to our system, the Dragon toolkit (cf. C.3), used for the generation of the lemma of the words, one of the features under consideration by our CBR classifiers.

After the processing of the input files, the output file is handled by the U-Compare framework and the results are presented in a graphical interface which allows the visualization of the given proteins and the entities, events and modifiers that have been extracted by the Moara BioEvent Extractor, as showed in Figure F.3.

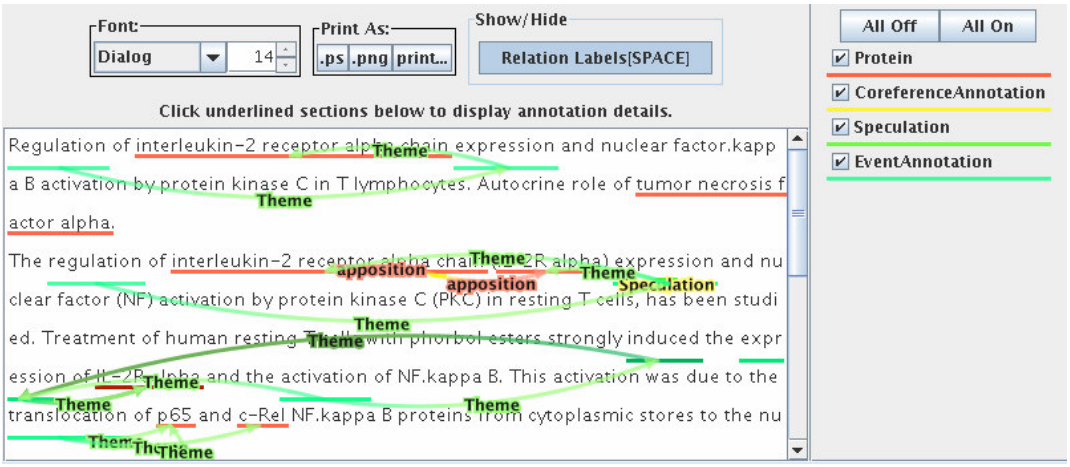


Figure F.3: Example of the results in the U-Compare framework.
The proteins are shown underlined (in red), as well as the event triggers (in green). The association of the event triggers to the arguments (e.g., Theme) is represented as a green curved line. Modifiers, such as speculation, are shown in yellow.

Tool name	Paper link	Tasks	Open-source	Availability	Training NER	Training EMN	Curated dictionary or rules for EMN	Integrations, Frameworks
Moara	(Neves, Carazo et al. 2010)	NER, EMN	✓	Java library	✓	✓		U-Compare (NER)
AbGene	(Tanabe and Wilbur 2002)	NER		Binaries				
ABNER	(Settles 2005)	NER	✓	Java library	✓			U-Compare, Whatizit
BANNER	(Leaman and Gonzalez 2008)	NER	✓	Web application, Java library	✓			
GNAT	(Hakenberg, Plake et al. 2008)	NER, EMN		Web application		✓	✓	
GENIA Tagger	(Tsuruoka, Tateishi et al. 2005)	NER	✓	Web application	✓			U-Compare
ProMiner	(Hanisch, Fundel et al. 2005)	NER, EMN		Commercial use			✓	
Whatizit	(Rebholz-Schuhmann, Arregui et al. 2008)	NER, EMN		Web application, Web service				

Table F.1: Comparison of the available tools for named-entity recognition and normalization.
The table shows a comparison among any available systems. NER stands for "named-entity recognition" and "EMN" for entity mention normalization.

APPENDIX G: PUBLICATIONS RELATED TO THE THESIS

G.1 Journals

Kano Y, Björne J, Ginter F, Buyko E, Hahn U, Cohen K B, Verspoor K, Roedor C, Hunter L E, Kilicoglu H, Bergler S, Van Landeghem S, Van Parys T, Van de Peer Y, Miwa M, Ananiadou S, **Neves M**, Pascual-Montano A, Özgür A, Radev D R, Riedel S, Sætre R, Chun H-W, Kim J-D, Pyysalo S, Ohta T, and Tsujii J. **U-Compare Bio-Event Meta-Service: Compatible BioNLP Event Extraction Services**, *BMC Bioinformatics*, 2011, 12:481.

Rebholz-Schuhmann D, Jimeno A, Li Ch, Kafkas S, Lewin I, Kang N, Corbett P, Milward D, Buyko E, Beisswanger E, Hornbostel K, Kouznetsov A, Witte R, Laurila J B, Baker Ch JO, Kuo Ch, Clematide S, Rinaldi F, Farkas R, Móra G, Hara K, Furlong L, Rautschka K, **Neves M L**, Pascual-Montano A, Wei Q, Collier N, Mahbub Chowdhury F, Lavelli A, Berlanga R, Morante R, Van Asch V, Daelemans W, Marina J L, van Mulligen E, Kors J and Hahn U. **Assessment of NER solutions against the first and second CALBC Silver Standard Corpus**, *SMBM 2010 special issue in the Journal of Biomedical Semantics*, 2011, 2(Suppl 5):S11.

Neves M L, Carazo J M and Pascual-Montano A. Moara: a Java library for extracting and normalizing gene and protein mentions, *BMC Bioinformatics*, 2010, 11:157.

Neves M, Carazo J M and Pascual-Montano A. **Extracting and normalizing gene/protein mentions with the flexible and trainable Moara Java library**, *BioLINK Special Interest Group, ISBM/ECCB 2009, Lecture Notes on Bioinformatics: Linking, Literature, Information, and Knowledge for Biology*, 6004, pp. 71--80. Springer, Heidelberg, 2010.

Smith L, Tanabe L K, Ando R J, Kuo C-J, Chung I-F, Hsu C-N, Lin Y-S, Klinger R, Friedrich C M, Ganchev K, Torii M, Liu H, Haddow B, Strubble C A, Povinelli R J, Vlachos A, Baumgartner W A, Hunter L, Carpenter B, Tsai R, Dai H-J, Liu F, Chen Y, Sun C, Katrenko S, Adriaans P, Blaschke C, Torres Perez R, **Neves M**, Nakov P, Divoli A, Mana M, Mata-Vazquez J and Wilbur W J. **Overview of BioCreative II Gene Mention Recognition**, *Genome Biology*, 2008, 9(Suppl 2):S2 (1 September 2008).

G.2 Conferences and Workshops

Rebholz-Schuhmann D, Jimeno A, Li Ch, Kafkas S, Lewin I, Kang N, Corbett P, Milward D, Buyko E, Beisswanger E, Hornbostel K, Kouznetsov A, Witte R, Laurila J B, Baker Ch JO, Kuo Ch, Clematide S, Rinaldi F, Farkas R, Móra G, Hara K, Furlong L, Rautschka K, **Neves M L**, Pascual-Montano A, Wei Q, Collier N, Mahbub Chowdhury F, Lavelli A, Berlanga R, Morante R, Van Asch V, Daelemans W, Marina J L, van Mulligen E, Kors J and Hahn U. **Assessment of NER solutions against the first and second CALBC Silver Standard Corpus**, *Proceedings of the 4th International Symposium on Semantic Mining in Biomedicine (SMBM'10)*, pages 66-74, 25th-26th October 2010, European Bioinformatics Institute, Hinxton, Cambridgeshire, United Kingdom.

Neves M, Marina J L and Pascual-Montano A. **BioLabeler and Moara in the First Round of the CALBC challenge**, *Proceedings of the First CALBC Workshop*, June 17th-18th, European Bioinformatics Institute, Hinxton, Cambridgeshire, United Kindom.

Neves M, Carazo J M and Pascual-Montano A. **Moara project: a flexible and trainable Java library for the gene/protein recognition and normalization tasks**, *Proceedings of the ISMB BioLINK'09*, pp. 241-244, June 28th 2009, Stockholm, Sweden.

Neves M, Carazo J M and Pascual-Montano A. **Extraction of biomedical events using case-based reasoning**, *Proceedings of the BioNLP'09 Shared Task on Event Extraction Workshop at NAACL-HLT 2009*, pp. 68-76, June 5th 2009, Boulder, CO, USA.

Neves M, Chagoyen M, Carazo J M and Pascual-Montano A. **CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem**, *Proceedings of the BioNLP 2008 Workshop at ACL 2008*, pp. 108-109, June 2008, Columbus, OH, USA.

Neves M, Chagoyen M, Carazo J M and Pascual-Montano A. **A new methodology for gene normalization using a mix of taggers, global alignment matching and document similarity disambiguation**, *VIII Jornadas de Bioinformatica*, Valencia, Spain, 2008.

Neves M L. **Identifying Gene Mentions by Case-Based Classification**, *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, pp. 77-79, Madrid, Spain, 2007.

BIBLIOGRAPHY

(2009). "The Universal Protein Resource (UniProt) 2009." Nucleic Acids Res **37**(Database issue): D169-174.

Aamodt, A. and E. Plaza (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." AI Communications **7**(1): 39-59.

Ananiadou, S. and G. Nenadic (2006). Automatic Terminology Management in Biomedicine. Text Mining for Biology and Biomedicine. S. Ananiadou and J. McNaught, Artech House**:** 67-97.

Ananiadou, S., S. Pyysalo, et al. "Event extraction for systems biology by text mining the literature." Trends Biotechnol.

Ando, R. K. (2007). BioCreative II Gene Mention Tagging System at IBM Watson. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Ashburner, M., C. A. Ball, et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nat Genet **25**(1): 25-29.

Baumgartner, W. A., Jr., H. L. Johnson, et al. (2007). An integrated approach to concept recognition I biomedical text. Proceedings of the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Baumgartner, W. A., Jr., Z. Lu, et al. (2008). "Concept recognition for extracting protein interaction relations from biomedical text." Genome Biol **9 Suppl 2**: S9.

Bjorne, J., J. Heimonen, et al. (2009). Extracting Complex Biological Events with Rich Graph-Based Features Sets. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop, Boulder, CO, USA.

Bundschus, M., M. Dejori, et al. (2008). "Extraction of semantic biomedical relations from text using conditional random fields." BMC Bioinformatics **9**: 207.

Cohen, A. M. (2007). Automatically Expanded Dictionaries with Exclusion Rules and Support Vector Machines Text Classifiers: Approaches to the BioCreative 2 GN and PPI-IAS Tasks. Proceedings of the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Cohen, W. C., P. Ravikumar, et al. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. IIWeb Workshop on International Joint Conference on Artificial Intelligence, Acapulco, Mexico.

Crim, J., R. McDonald, et al. (2005). "Automatically annotating documents with normalized gene lists." BMC Bioinformatics **6 Suppl 1**: S13.

Chapman, W. W. and K. B. Cohen (2009). "Current issues in biomedical text mining and natural language processing." J Biomed Inform **42**(5): 757-759.

Chatr-aryamontri, A., A. Ceol, et al. (2007). "MINT: the Molecular INTeraction database." Nucleic Acids Res **35**(Database issue): D572-574.

Chen, Y., F. Liu, et al. (2007). Gene Mention Recognition Using Lexicon Match Based Two-Layer Support Vector Machines. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Cherry, J. M., C. Adler, et al. (1998). "SGD: Saccharomyces Genome Database." Nucleic Acids Res **26**(1): 73-79.

Daelemans, W., J. Zavrel, et al. (1996). MBT: A Memory-Based Part of Speech Tagger-Generator. Fourth Workshop on Very Large Corpora, Copenhagen, Denmark.

Dai, H.-J., H.-C. Hung, et al. (2007). IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-Task. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

de Marneffe, M.-C., B. MacCartney, et al. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. Language Resources and Evaluation (LREC), Genoa, Italy.

Eppig, J. T., C. J. Bult, et al. (2005). "The Mouse Genome Database (MGD): from genes to mice--a community resource for mouse biology." Nucleic Acids Res **33**(Database issue): D471-475.

Farkas, R. (2008). "The strength of co-authorship in gene name disambiguation." BMC Bioinformatics **9**: 69.

Faro, A., D. Giordano, et al. (2011). "Combining literature text mining with microarray data: advances for system biology modeling." Brief Bioinform.

Fenton, S. and M. Williams (2005). "Getting to know PubMed: an overview." J Ahima **76**(3): 60A-60D.

Finkel, J., S. Dingare, et al. (2005). "Exploring the boundaries: gene and protein identification in biomedical text." BMC Bioinformatics **6 Suppl 1**: S5.

Fukuda, K., T. Tsunoda, et al. (1998). Toward Information Extraction: Identifying protein names from biological papers Pacific Symposium on Biocomputing (PSB98), Hawaii, USA.

Fundel, K., D. Guttler, et al. (2005). "A simple approach for protein name identification: prospects and limits." BMC Bioinformatics **6 Suppl 1**: S15.

Fundel, K., R. Kuffner, et al. (2007). "RelEx--relation extraction using dependency parse trees." Bioinformatics **23**(3): 365-371.

Ganchev, K., K. Crammer, et al. (2007). Penn/UMass/CHOP BioCreative II systems. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

García, F. C., E. Puertas, et al. (2007). Attribute Analysis in Biomedical Text Classification. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Gelbart, W. M., M. Crosby, et al. (1997). "FlyBase: a Drosophila database. The FlyBase consortium." Nucleic Acids Res **25**(1): 63-66.

Giuliano, C., A. Lavelli, et al. (2006). Exploiting Shallow Linguistic Information for Relation Extraction From Biomedical Literature. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006). Trento, Italy.

Hahn, U. and J. Wermter (2006). Levels of Natural Language Processing for Text Mining. Text Mining for Biology and Biomedicine. S. Ananiadou and J. McNaught, Artech House**:** 13-41.

Hakenberg, J., C. Plake, et al. (2008). "Inter-species normalization of gene mentions with GNAT." Bioinformatics **24**(16): i126-132.

Hanisch, D., K. Fundel, et al. (2005). "ProMiner: rule-based protein and gene entity recognition." BMC Bioinformatics **6 Suppl 1**: S14.

Hersh, W. (2008). Information Retrieval: A Health and Biomedical Perspective. New York, Springer New York.

Hirschman, L., M. Colosimo, et al. (2005). "Overview of BioCreAtIvE task 1B: normalized gene lists." BMC Bioinformatics **6 Suppl 1**: S11.

Holloway, A. J., R. K. van Laar, et al. (2002). "Options available--from start to finish--for obtaining data from DNA microarrays II." Nat Genet **32 Suppl**: 481-489.

Huang, H. S., Y.-S. Lin, et al. (2007). High-Recall Gene Mention Recognition by Unification of Multiple Backward Parsing Models. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Jackson, P. and I. Moulinier (2002). Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization, John Benjamins Publishing Company.

Jenssen, T. K., A. Laegreid, et al. (2001). "A literature network of human genes for high-throughput analysis of gene expression." Nat Genet **28**(1): 21-28.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany.

Kano, Y., W. A. Baumgartner, Jr., et al. (2009). "U-Compare: share and compare text mining tools with UIMA." Bioinformatics.

Kano, Y., J. Bjorne, et al. (2011). "U-Compare bio-event meta-service: compatible BioNLP event extraction services." BMC Bioinformatics **12**(1): 481.

Katrenko, S. and P. W. Adriaans (2007). Using Semi-Supervised Techniques to Detect Gene Mentions. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Kerrien, S., Y. Alam-Faruque, et al. (2007). "IntAct--open source resource for molecular interaction data." Nucleic Acids Res **35**(Database issue): D561-565.

Kilicoglu, H. and S. Bergler (2009). Syntactic Dependency Based Heuristics for Biological Event Extraction. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop. Boulder, CO, USA.

Kim, J.-D., T. Ohta, et al. (2009). Overview of BioNLP'09 Shared Task on Event Extraction. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop, Boulder, CO, USA.

Kim, J. D., T. Ohta, et al. (2008). "Corpus annotation for mining biomedical events from literature." BMC Bioinformatics **9**: 10.

Klein, D. and C. D. Manning (2003). Accurate Unlexicalized Parsing. 41st Meeting of the Association for Computational Linguistics.

Klinger, R., C. M. Friedrich, et al. (2007). Named Entity Recognition with Combinations of Conditional Random Fields. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Koike, A. and T. Takagi (2004). Gene/Protein/family name recognition in biomedical literature. BioLINK 2004 - Biological Literature, Ontologies and Databases: Tools for Users, Boston, USA.

Kolodner, J. (1992). "An Introduction to Case-Based Reasoning." Artificial Intelligence Review **6**: 3-34.

Krallinger, M., F. Leitner, et al. (2008). "Overview of the protein-protein interaction annotation extraction task of BioCreative II." Genome Biol **9 Suppl 2**: S4.

Krallinger, M., A. Morgan, et al. (2008). "Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge." Genome Biol **9 Suppl 2**: S1.

Krauthammer, M., A. Rzhetsky, et al. (2000). "Using BLAST for identifying gene and protein names in journal articles." Gene **259**(1-2): 245-252.

Kuo, C.-J., Y.-M. Chang, et al. (2007). Rich Feature Set, Unification of Bidirectional Parsing and Dictionary Filtering for High F-Score Gene Mention Tagging. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Lafferty, J., A. McCallum, et al. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 18th International Conference on Machine Learning, San Francisco, CA, Morgan Kaufmann.

Lau, W. W. and C. A. Johnson (2007). Rule-Based Gene Normalization with a Statistical and Heuristic Confidence Measure. Proceedings of the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Leaman, R. and G. Gonzalez (2008). "BANNER: an executable survey of advances in biomedical named entity recognition." Pac Symp Biocomput: 652-663.

Leser, U. and J. Hakenberg (2005). "What makes a gene name? Named entity recognition in the biomedical literature." Brief Bioinform **6**(4): 357-369.

Liu, H., C. Wu, et al. (2004). BioTagger: A Biological Entity Tagging System. BioCreAtIvE Workshop Handouts, Granada, Spain.

Lodish, H., A. Berk, et al. (2000). Molecular Cell Biology. New York, W. H. Freeman.

MacBeath, G. (2002). "Protein microarrays and proteomics." Nat Genet **32 Suppl**: 526-532.

Maglott, D., J. Ostell, et al. (2007). "Entrez Gene: gene-centered information at NCBI." Nucleic Acids Res **35**(Database issue): D26-31.

Marcus, M. P., B. Santorini, et al. (1993). "Building a Large Annotated Corpus of English: The Penn TreeBank." Computational Linguistics **19**(2): 313-330.

Miyao, Y., R. Saetre, et al. (2008). Task-Oriented Evaluation of Syntactic Parsers and Their Representations. 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08:HLT), Columbus, USA.

Miyao, Y., K. Sagae, et al. (2009). "Evaluating contributions of natural language parsers to protein-protein interaction extraction." Bioinformatics **25**(3): 394-400.

Móra, G., R. Farkas, et al. (2009). Exploring ways beyond the simple supervised learning approach for biological event extraction. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop. Boulder, CO, USA.

Morante, R., V. Van Asch, et al. (2009). A memory-based learning approach to event extraction in biomedical texts. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop, Boulder, CO, USA.

Morgan, A. A., Z. Lu, et al. (2008). "Overview of BioCreative II gene normalization." Genome Biology **9 Suppl 2**: S3.

Neves, M. (2007). Identifying Gene Mentions by Case-Based Reasoning. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Neves, M., J. M. Carazo, et al. (2009). Extraction of biomedical events using case-based reasoning. Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop, Boulder, CO, USA.

Neves, M., M. Chagoyen, et al. (2008). CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem. BioNLP 2008 Workshop at ACL 2008, Columbus, OH, USA.

Neves, M., M. Chagoyen, et al. (2008). A new methodology for gene normalization using a mix of taggers, global alignment matching and document similarity disambiguation. Jornadas de Bioinformática, Valencia, Spain.

Neves, M. L., J. M. Carazo, et al. (2010). "Moara: a Java library for extracting and normalizing gene and protein mentions." BMC Bioinformatics **11**(1): 157.

Ohta, T., Y. Tateishi, et al. (2002). The Genia corpus: an annotated research abstract corpus in molecular biology domain. Human Language Technology Conference (HTL 2002), San Diego, USA.

Park, J. C. and J.-j. Kim (2006). Named Entity Recognition. Text Mining for Biology and Biomedicine. S. Ananiadou and J. McNaught, Artech House**:** 121-142.

Porter, M. (1980). "An algorithm for suffix stripping." Program **14**(3): 130-137.

Povey, S., R. Lovering, et al. (2001). "The HUGO Gene Nomenclature Committee (HGNC)." Hum Genet **109**(6): 678-680.

Pyysalo, S., A. Airola, et al. (2008). "Comparative analysis of five protein-protein interaction corpora." BMC Bioinformatics **9**(Suppl 3): S6.

Rebholz-Schuhmann, D., M. Arregui, et al. (2008). "Text processing through Web services: calling Whatizit." Bioinformatics **24**(2): 296-298.

Rosario, B. and M. A. Hearst (2004). Classifying Semantic Relations in Bioscience Text. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), Barcelona, Spain.

Salton, G. and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval." Information Processing & Management **24**(5): 513-523.

Schuemie, M., R. Jelier, et al. (2007). Peregrine: Lightweight gene name normalization by dictionary lookup. Proceedings of the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Schwartz, A. and M. Hearst (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. Proceedings of the Pacific Symposium on Biocomputing, Kauai, USA.

Settles, B. (2005). "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text." Bioinformatics **21**(14): 3191-3192.

Shang, H. and T. H. Merrettal (1996). "Tries for Approximate String Matching." IEEE Transactions on Knowledge and Data Engineering **8**(4): 540-547.

Shatkay, H. and R. Feldman (2003). "Mining the biomedical literature in the genomic era: an overview." J Comput Biol **10**(6): 821-855.

Slade, S. (1991). "Case-Based Reasoning: A Research Paradigm." AI Magazine **12**(1): 42-55.

Smith, L., L. K. Tanabe, et al. (2008). "Overview of BioCreative II gene mention recognition." Genome Biology **9 Suppl 2**: S2.

Smith, L. H., L. Tanabe, et al. (2005). MedTag: A Collection of Biomedical Annotations. Joint ACL Workshop and BioLINK SIG (ISMB) on Linking Biological Literature Ontologies and Databases

Spasic, I. and S. Ananiadou (2005). "A flexible measure of contextual similarity for biomedical terms." Pac Symp Biocomput: 197-208.

Spasic, I., S. Ananiadou, et al. (2005). "MaSTerClass: a case-based reasoning system for the classification of biomedical terms." Bioinformatics **21**(11): 2748-2758.

Struble, C. A., R. J. Povinelli, et al. (2007). Combined Conditional Random Fields and n-Gram Language Models for Gene Mention Recognition. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Tamames, J. (2005). "Text detective: a rule-based system for gene annotation in biomedical texts." BMC Bioinformatics **6 Suppl 1**: S10.

Tamames, J. and A. Valencia (2006). "The success (or not) of HUGO nomenclature." Genome Biol **7**(5): 402.

Tanabe, L. and W. J. Wilbur (2002). "Tagging gene and protein names in biomedical text." Bioinformatics **18**(8): 1124-1132.

Tanabe, L., N. Xie, et al. (2005). "GENETAG: a tagged corpus for gene/protein named entity recognition." BMC Bioinformatics **6 Suppl 1**: S3.

Thompson, P., S. A. Iqbal, et al. (2009). "Construction of an annotated corpus to support biomedical information extraction." BMC Bioinformatics **10**: 349.

Tikk, D., P. Thomas, et al. (2010). "A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature." PLoS Comput Biol **6**: e1000837.

Tjong Kim Sang, E. F. and J. Veenstra (1999). Representing text chunks. Ninth conference on European chapter of the Association for Computational Linguistics, Bergen, Norway, Association for Computational Linguistics.

Tsuruoka, Y., J. McNaught, et al. (2007). "Learning string similarity measures for gene/protein name dictionary look-up using logistic regression." Bioinformatics **23**(20): 2768-2774.

Tsuruoka, Y., Y. Tateishi, et al. (2005). Developing a Robust Part-of-Speech Tagger for Biomedical Text. Advances in Informatics - 10th Panhellenic Conference on Informatics, LNCS.

Tsuruoka, Y. and J. Tsujii (2003). Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. ACL-03 Workshop on Natural Language Processing in Biomedicine, Sapporo, Japan.

Tsuruoka, Y., J. Tsujii, et al. (2008). "FACTA: a text search engine for finding associated biomedical concepts." Bioinformatics **24**(21): 2559-2560.

Tsuruoka, Y. and J. i. Tsujii (2005). Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. HLT/EMNLP - Conference on Empirical Methods

in Natural Language Processing, Human Language Technology Conference Vancouver, Canada.

Vlachos, A. (2007). Tackling the BioCreative2 Gene Mention task with Conditional Random Fields and Syntactic Parsing. Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

Weber, R. O., K. D. Ashley, et al. (2005). "Textual case-based reasoning." Knowl. Eng. Rev. **20**(3): 255-260.

Wermter, J., K. Tomanek, et al. (2009). "High-performance gene name normalization with GeNo." Bioinformatics **25**(6): 815-821.

Witten, I. H. and E. Frank (2005). Data mining: Practical machine learning tools and techniques. San Francisco, Morgan Kaufmann.

Yakushiji, A., Y. Tateisi, et al. (2001). "Event extraction from biomedical papers using a full parser." Pac Symp Biocomput: 408-419.

Yeh, A., A. Morgan, et al. (2005). "BioCreAtIvE task 1A: gene mention finding evaluation." BMC Bioinformatics **6 Suppl 1**: S2.

Zhou, D. and Y. He (2008). "Extracting interactions between proteins from the literature." J Biomed Inform **41**(2): 393-407.

Zhou, G., D. Shen, et al. (2005). "Recognition of protein/gene names from text using an ensemble of classifiers." BMC Bioinformatics **6 Suppl 1**: S7.

Zhou, X., X. Zhang, et al. (2007). Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining. 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Patras, Greece.